



Developing and deploying enterprise GIS solutions with ArcGIS Server and Explorer

Timothy R. Weisenburger
ESRI

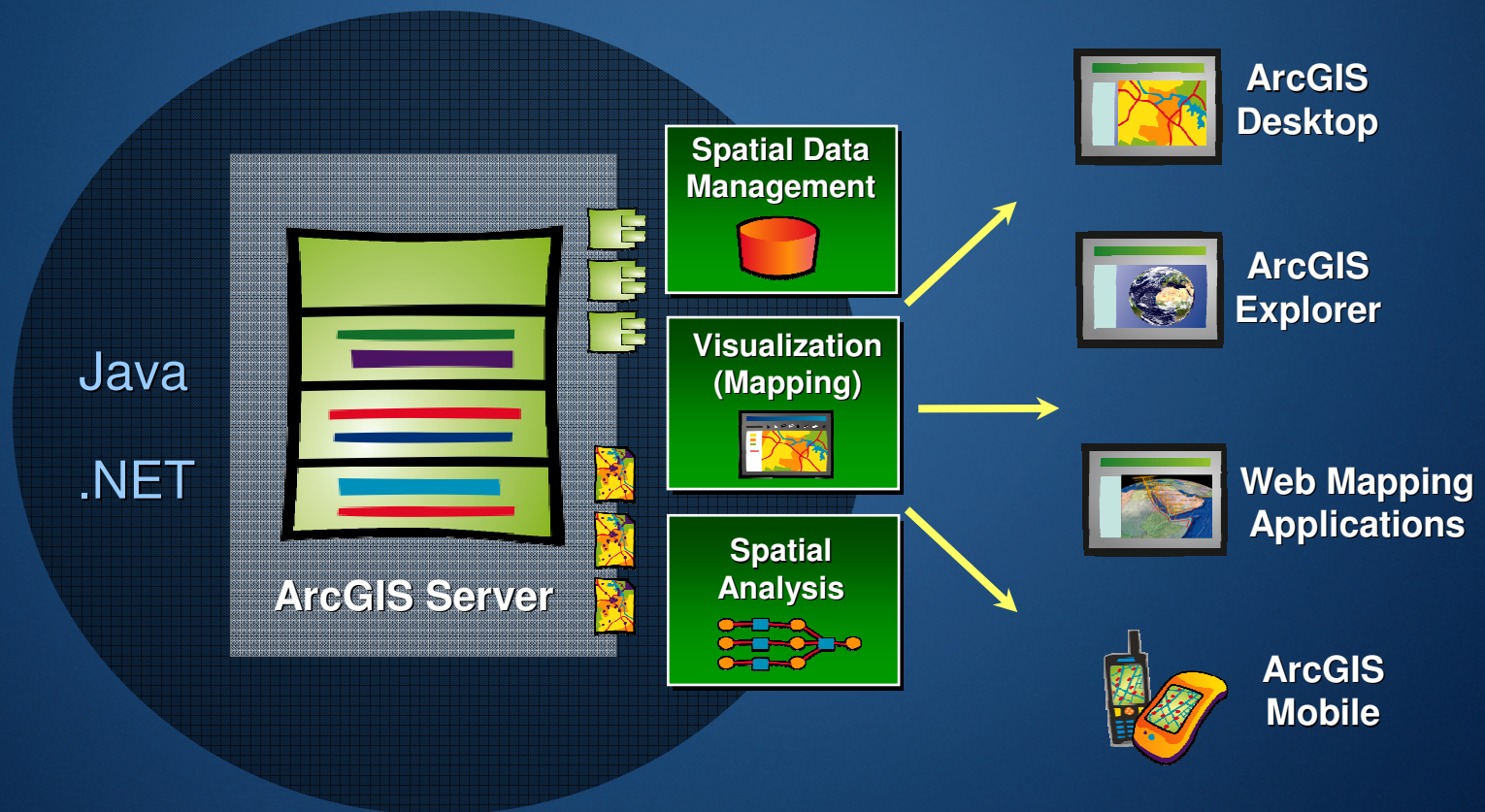
Agenda

Introduction ArcGIS Server Architecture

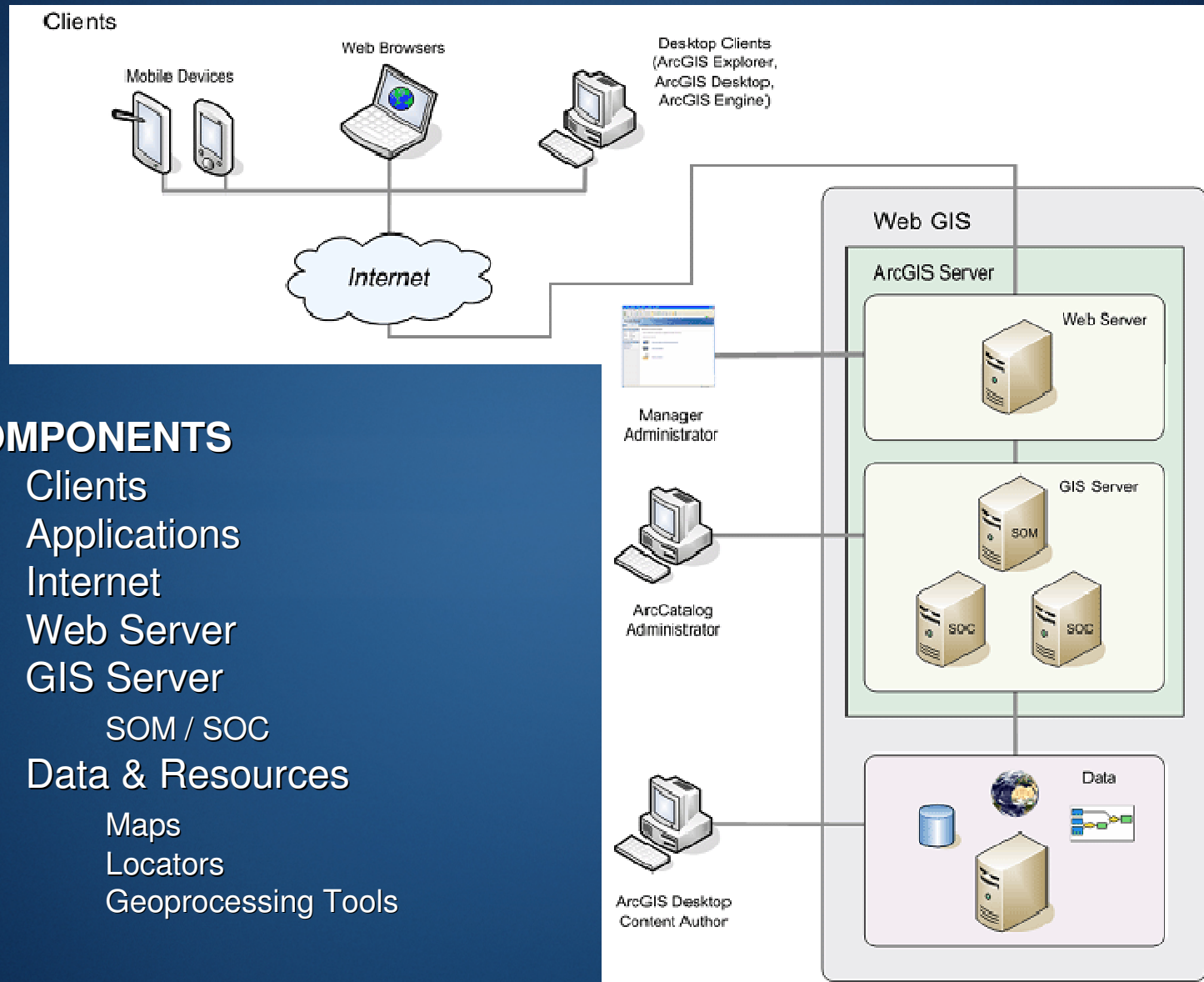
- Tuning and Configuration of Services
- Map Document Optimization
 - Dynamic Map Services
 - Map Caching
- Developer Options in .NET
 - ArcGIS Server Manager
 - The Web ADF
 - Non WebADF SOAP API
 - Help Info
- ArcGIS Explorer
 - Introduction
 - Customization and Configuration
 - Custom Tasks – Geoprocessing and the SDK

ArcGIS Server 9.2

- Complete & Integrated server-based GIS
- Out-of-the-box applications and services
- Tremendous developer opportunities



Architecture of ArcGIS Server



COMPONENTS

- Clients
- Applications
- Internet
- Web Server
- GIS Server
 - SOM / SOC
- Data & Resources
 - Maps
 - Locators
 - Geoprocessing Tools

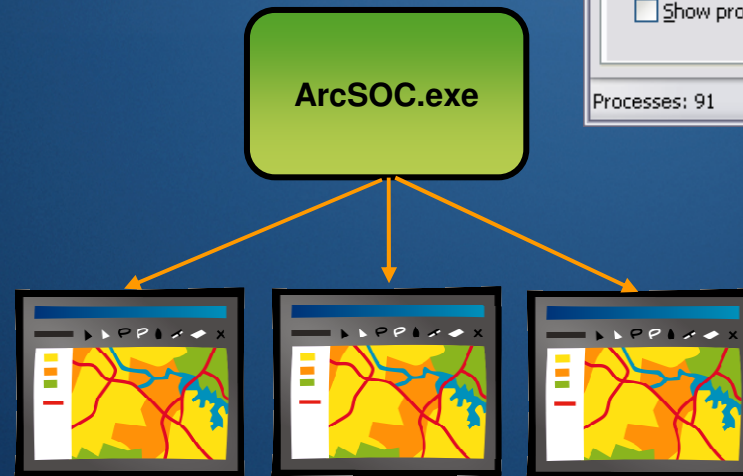


Tuning and Configuration of Services

Meet the Benchmark Expectation

Service Instances, Processes and Threads

- Service Instance – A single occurrence of a service that represents an application (MXD)
- Thread – Equates to a service instance at the operating system level
- Instances are hosted by ArcSOC processes



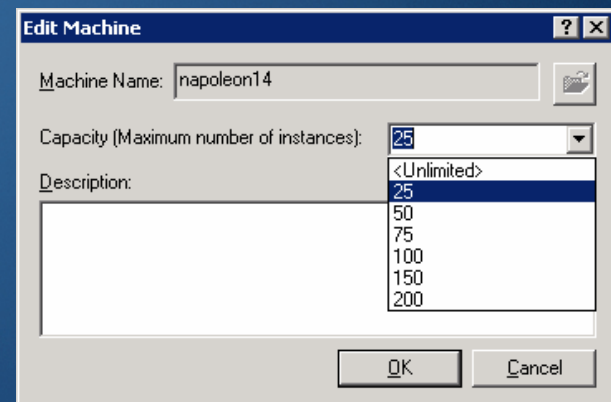
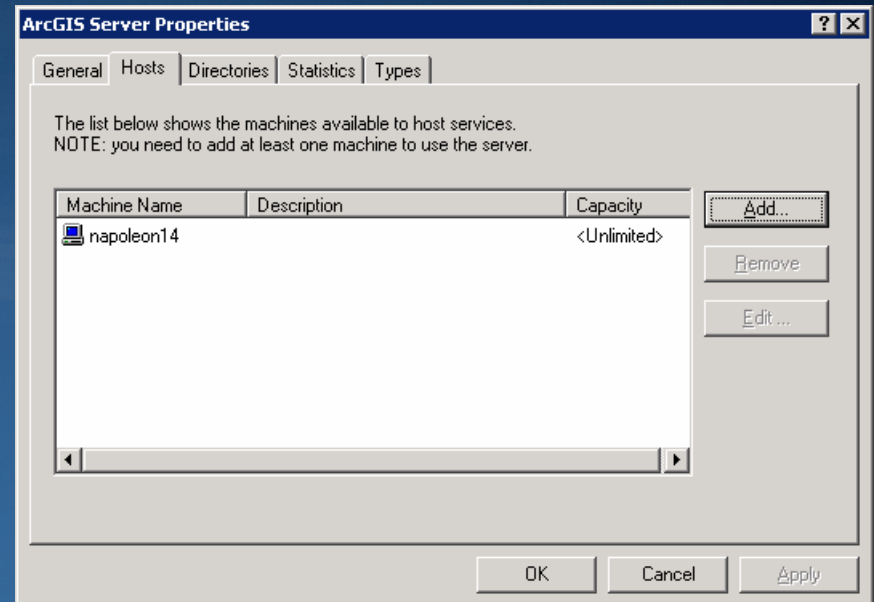
A screenshot of the Windows Task Manager window, specifically the "Processes" tab. The window title is "Windows Task Manager". The menu bar includes "File", "Options", "View", and "Help". The tabs are "Applications", "Processes", "Performance", and "Networking". The "Processes" tab is active, displaying a list of running processes with columns for "Image Name", "User Name", "CPU", and "Mem Usage".

Image Name	User Name	CPU	Mem Usage
ADU.exe	tweisenburger	00	4,648 K
alg.exe	LOCAL SERVICE	00	3,652 K
apdproxy.exe	tweisenburger	00	5,044 K
ApntEx.exe	tweisenburger	00	2,776 K
Apoint.exe	tweisenburger	00	5,072 K
AppROT.exe	tweisenburger	00	4,356 K
ArcCatalog.exe	tweisenburger	00	27,208 K
ARCGIS.EXE	SYSTEM	00	6,576 K
ArcSOC.exe	tweisenburger	00	5,032 K
ArcSOC.exe	tweisenburger	00	5,140 K
ArcSOC.exe	tweisenburger	00	48,768 K
ArcSOM.exe	tweisenburger	00	7,456 K
BAsfIpM.exe	SYSTEM	00	3,388 K
ccApp.exe	tweisenburger	00	7,772 K
ccEvtMgr.exe	SYSTEM	00	3,148 K
CcmExec.exe	SYSTEM	00	19,032 K
ccs.exe	SYSTEM	00	4,180 K
ccSetMgr.exe	SYSTEM	00	4,572 K
csrss.exe	SYSTEM	00	4,808 K

At the bottom of the window, there is a checkbox labeled "Show processes from all users" and a button labeled "End Process". The status bar at the very bottom shows "Processes: 91", "CPU Usage: 5%", and "Commit Charge: 1267M / 3430M".

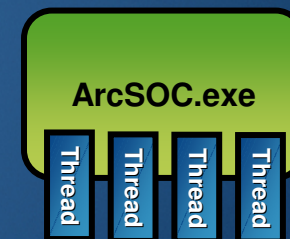
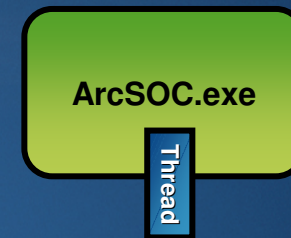
Setting Capacity

- Limits number of service instances running on a specific host machine.
- Once this limit is reached, Server starts replacing least recently used instances instead of creating new ones.
- Serve large number of services only part of which are used at any point in time
 - Supports “limited resource” scenario
 - Serving a large library of maps
 - Individual services rarely used

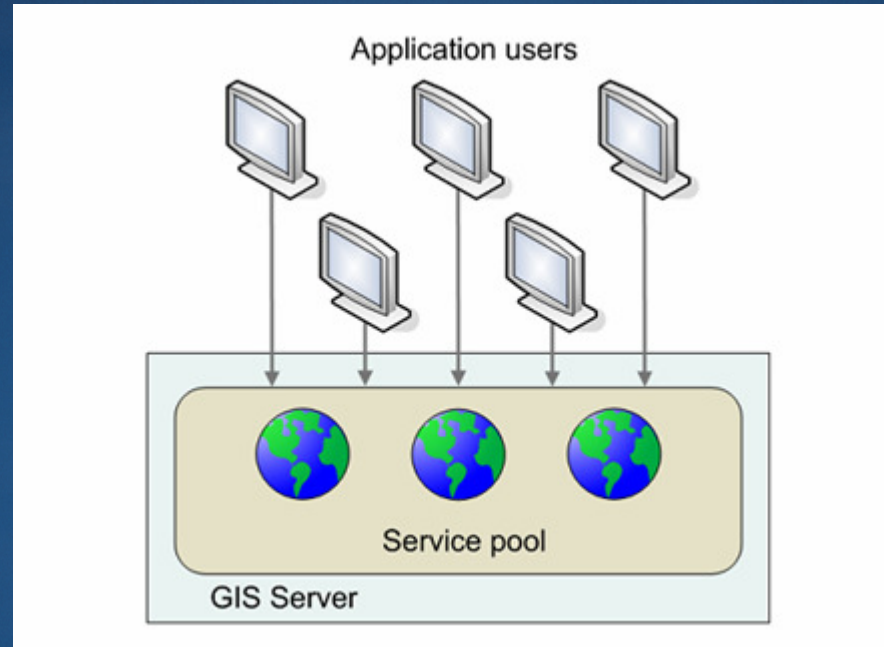


Isolation

- High Isolation Example: 12 service instances equates to 12 ArcSOC.exe processes with one instance/thread each
- Low Isolation Example: 12 service instances equates to 3 ArcSOC.exe processes with up to four instances/threads each
- Recommendation: Use high isolation
 - A failed instance is “isolated” to one ArcSOC.exe process

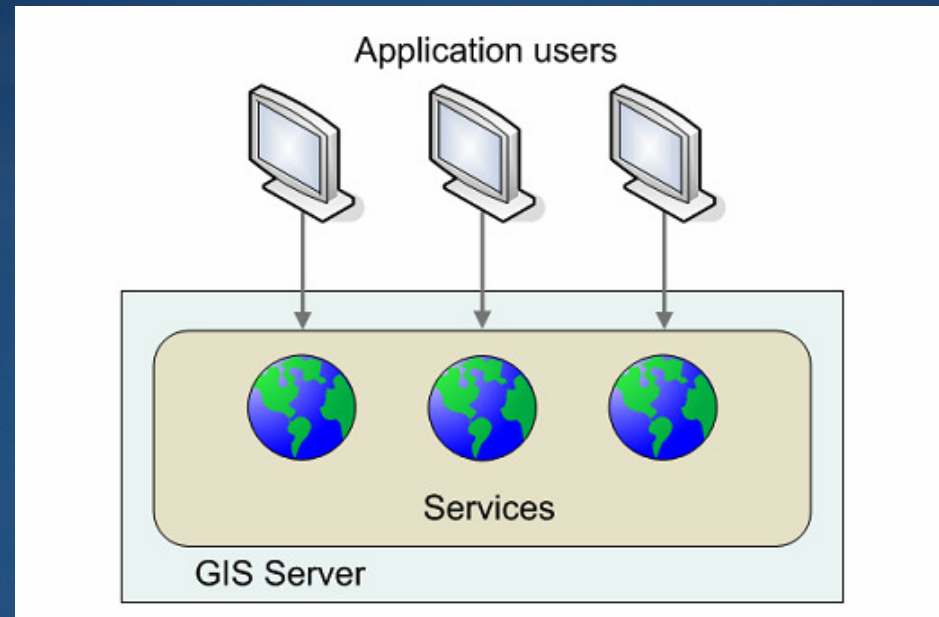


Pooled Service Model



- State information (e.g., Current extent, layer visibility, etc.) maintained in web server / browser
- Scales better due to shared object pool

Non-Pooled Service Model



- Typically holds its reference to the service for the duration of the application's session
- Number of users on the system can have no more than a 1:1 correlation with the number of running service instances
- Do not use internet connections to non-pooled

Configuring Pooled / Non-Pooled Instances

- Define Min-Max instances
- Typically 2-4 instances per SOC CPUs/cores
 - Depends on relative performance of data source
 - Depends on local / ArcSDE data
- Instances are distributed across all host servers

The screenshot shows the 'ArcGIS Server - Map Service Properties' dialog box with the 'Pooling' tab selected. The 'Pooling' section contains two radio buttons: 'Pooled - Used repeatedly by many clients.' (selected) and 'Not pooled - Used by a single client and disposed of after use.' Below these are two text boxes: 'Minimum number of instances:' with the value '1' and 'Maximum number of instances:' with the value '2'. A red circle highlights these two text boxes. The 'Timeouts' section contains two text boxes: 'The maximum time a client can use a service:' with the value '600' and 'The maximum time a client will wait to get a service:' with the value '60'. Both are followed by the unit 'seconds'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

ArcGIS Server - Map Service Properties

General Parameters Capabilities Pooling Processes

Pooling

This service should be:

☒ Pooled - Used repeatedly by many clients.

☐ Not pooled - Used by a single client and disposed of after use.

Minimum number of instances: 1

Maximum number of instances: 2

Timeouts

The maximum time a client can use a service: 600 seconds

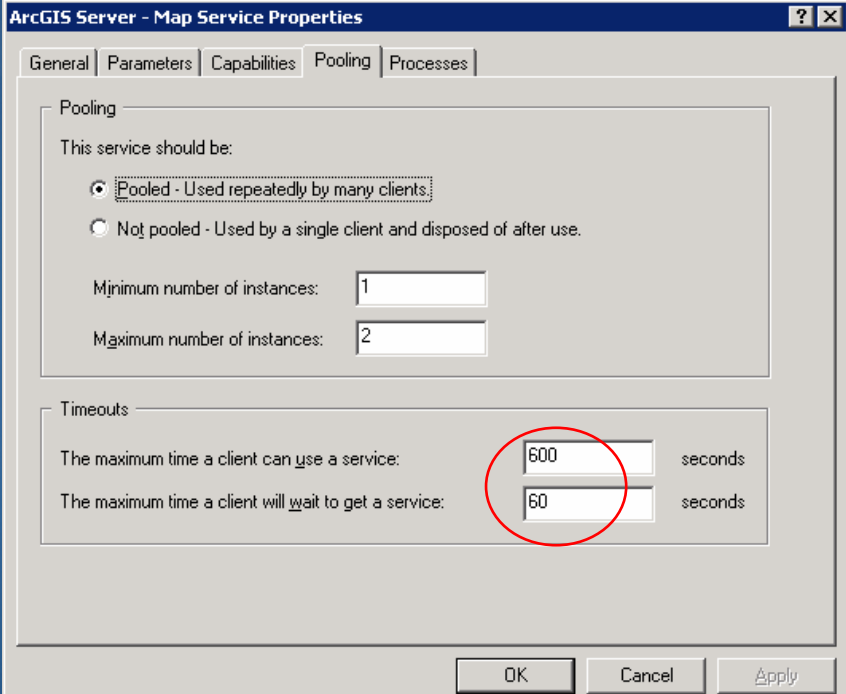
The maximum time a client will wait to get a service: 60 seconds

OK Cancel Apply

Wait Time and Usage Time

- Wait Time
 - Time it takes to Request a Service
 - Handle the Error
- Usage Time
 - How long can a service be used
 - Geoprocessing VS Mapping
 - Failure or Long Processes

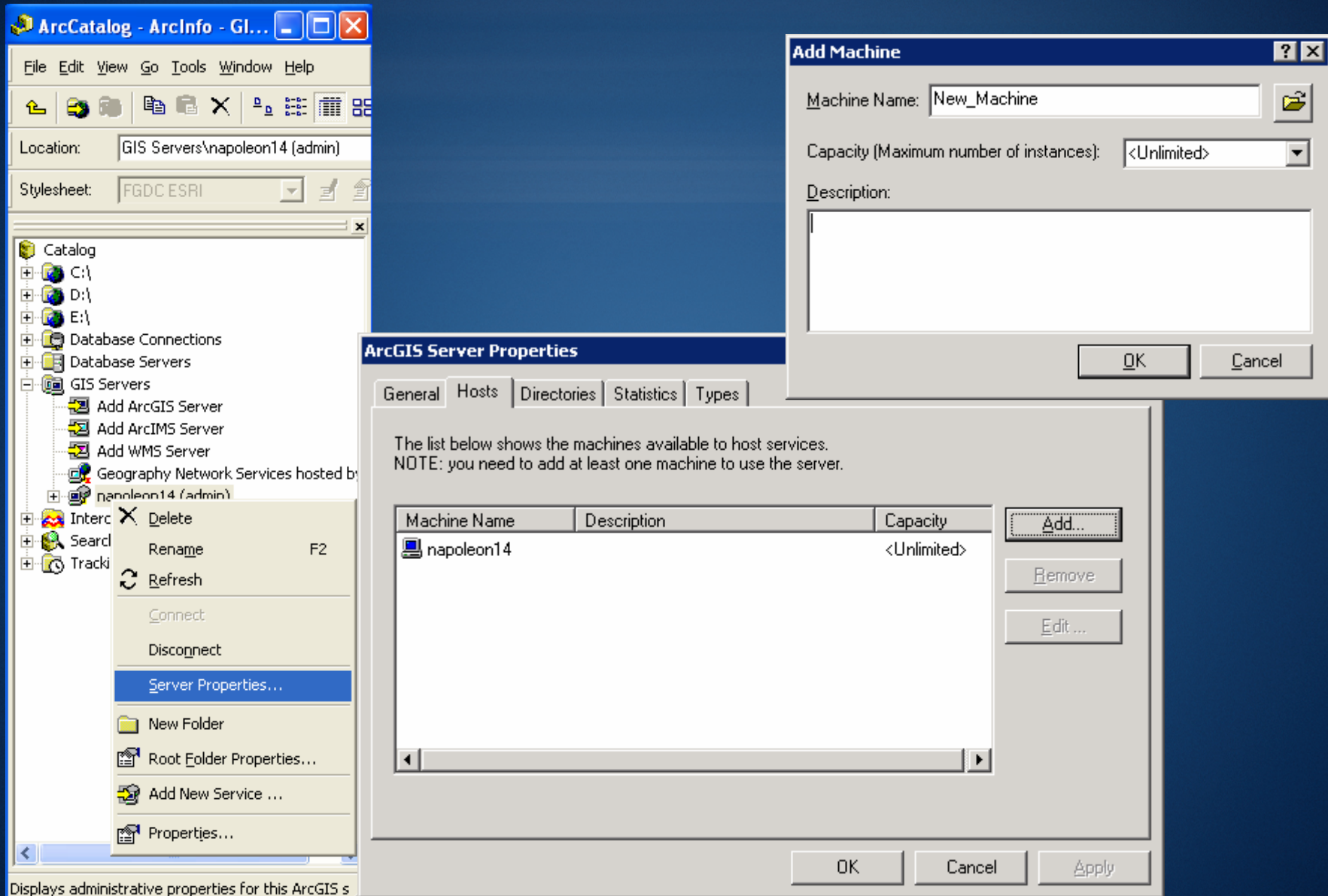
Statistics Tracks these Times



The screenshot shows the 'ArcGIS Server - Map Service Properties' dialog box with the 'Pooling' tab selected. The 'Pooling' section has two radio buttons: 'Pooled - Used repeatedly by many clients.' (selected) and 'Not pooled - Used by a single client and disposed of after use.' Below these are input fields for 'Minimum number of instances' (set to 1) and 'Maximum number of instances' (set to 2). The 'Timeouts' section has two input fields: 'The maximum time a client can use a service:' (set to 600) and 'The maximum time a client will wait to get a service:' (set to 60). Both 600 and 60 are circled in red. The dialog has 'OK', 'Cancel', and 'Apply' buttons at the bottom.

Property	Value	Unit
Minimum number of instances	1	
Maximum number of instances	2	
The maximum time a client can use a service	600	seconds
The maximum time a client will wait to get a service	60	seconds

Scaling Out – Adding More Computing Power





Dynamic Map Services

Best Practices – Map Services

- We encourage the use of cached map services and only when necessary use dynamic map services!

Considerations:

- Know your audience and their needs requirements
- Understand the trade-offs

Optimize Your Map Services - The MXD

- Maps can include both dynamic as well as static layers
- Design your maps for the Web –
 - Consideration with single source to multiple users
 - Minimize the data push
- Dynamic Layers = rapidly changing data
 - Roads symbolized by current snow depth
 - Electrical network showing the latest posted work order
 - Tracking Data / Updated GPS Point Data
- Static Layers = more slowly changing data
 - Landuse / Landcover
 - Road Network
 - Basemap data
 - Imagery
- Data Location and Access and Bandwidth Issues
 - SDE The Fastest
 - File based – Disk Contention
 - Don't Use Outside Services (ArcIMS, ArcGIS Server, WMS)
 - STAY WITH YOUR LOCAL RESOURCES
 - Outside resources are better utilized at the application level

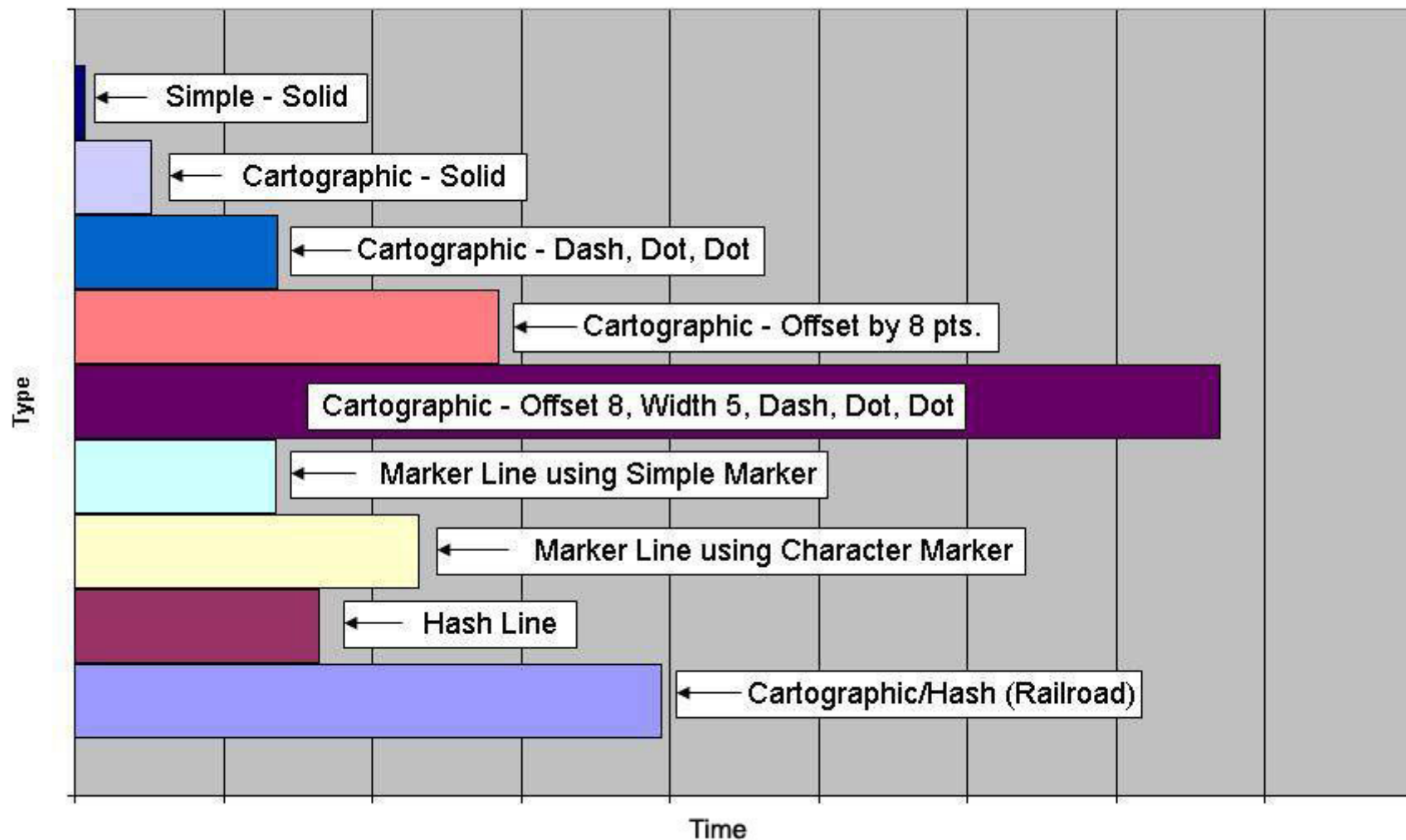
General Guidelines – Dynamic Maps

- Show relevant information
 - Start simple (additional layers can be toggled on by user)
 - Use field visibility (hide unnecessary attributes)
- Use scale dependencies
 - Use data appropriate for the given scale (generalize if necessary)
 - Display similar number of features at all scales for consistent user experience
- Remove Unused Layers and Data Frames
- Make sure data projections is the same as the Data Frame
- Use Definition Queries

Point, Line & Polygon rendering

- Points
 - Use single layer Simple or Character markers for best performance
 - Use EMF instead of bitmaps
 - Use Integer (vs. character) fields for symbol values
 - Avoid halos, complex shapes, masking
- Line & Polygons
 - Use ESRI_Optimized style
 - Outlines for all fills are simple lines instead of cartographic lines
 - Picture fills are EMF-based instead of BMP-based
 - Improves drawing performance by > 50%
 - Avoid cartographic lines (also includes polygon outline!)

Line Symbols



Text and Labeling

- Use annotation instead of labels
- Use indexed fields (reduce label SQL query number and complexity when possible)
- Use label and feature conflict weights sparingly
- Avoid special effects (fill patterns, halos, callouts, backgrounds)
- Avoid very large text size (60+ pts)
- Avoid Maplex for dynamic labeling
- Avoid overuse
- Avoid Highway Symbols
- Use Scale Dependencies

Mapping Application Image Size

- Map Request
 - 600 x 400



JPEG = 70 KB

- 1200 x 800



JPEG = 161 KB

- Recommendation:
 - Use reasonable output image size to support application while minimizing impact to network

Output Image Type

- Output image sizes vary by format and data type
 - 600 x 400 Example

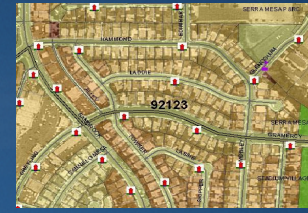
Raster



JPEG = 76 KB



PNG24 = 316 KB



BMP = 703 KB

Vector



JPEG = 70 KB



PNG24 = 30 KB



BMP = 703 KB

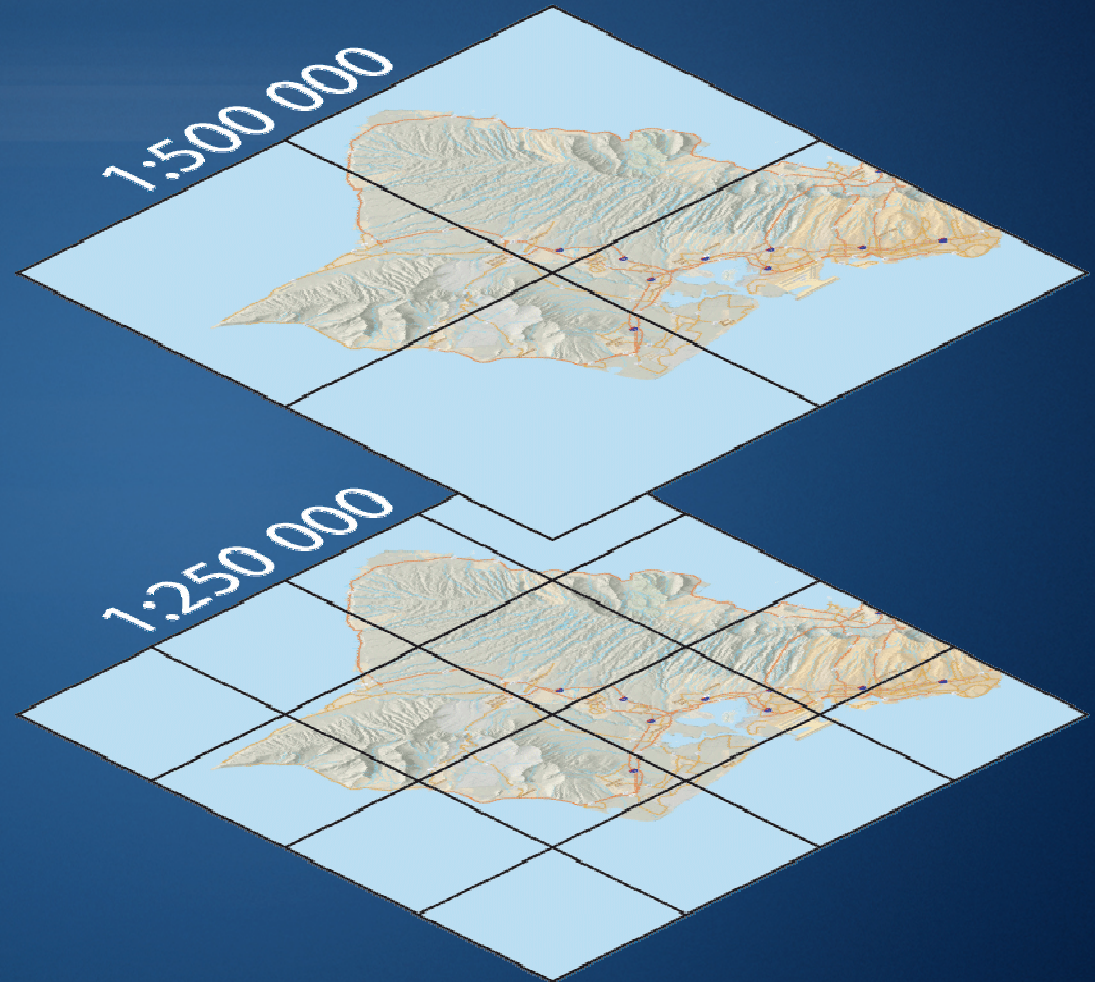
- Recommendation:
 - Use appropriate output type to support application while minimizing impact to network. Generally raster data is best served in a JPEG format, while vector data is best served in PNG format. Use PNG32 to support transparency.



Cached Map Services

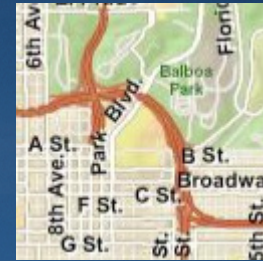
Cached Map Service

- Tiles pre-rendered at fixed scales
- Rapid display of static base maps
- Richer symbols and more information



What types of maps should I cache?

- Base maps
- Maps that don't change frequently
- Maps you won't be editing



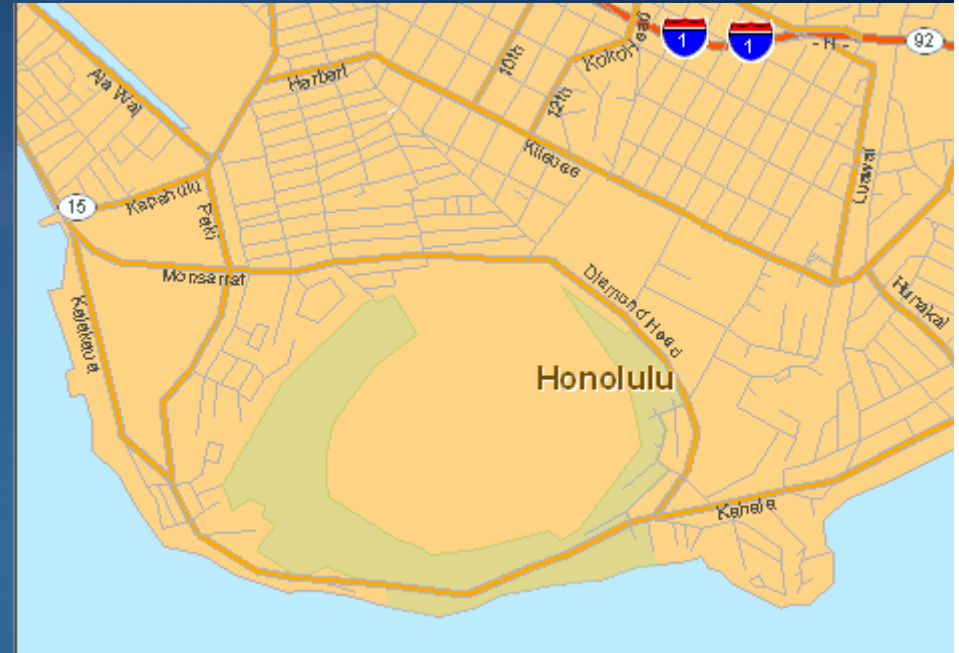
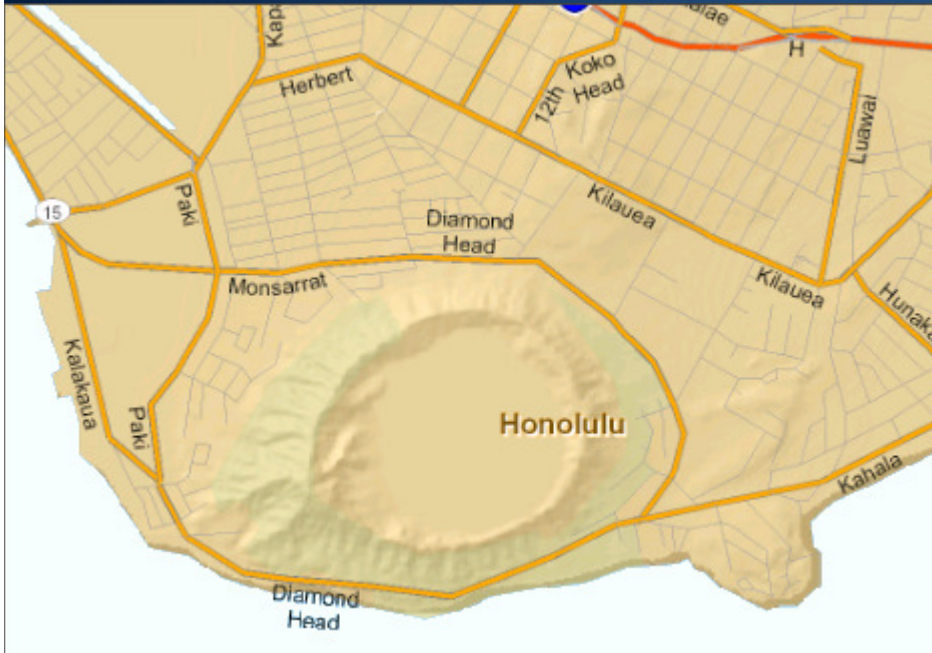
**As long as the Cache is available – Map Image requests are not required
Freeing up the Server Resources**

Classic Dynamic Mapping Trade-Off

Quality

vs.

Speed



- Shaded Relief
- Transparent Layers
- Maplex Labeling
 - 1.5 Seconds

- Low-res relief
- Solid colors
- Annotation
 - 4 Seconds

If you can cache your map, then there is no need to trade quality for performance!

Generating the map cache

- The caching process can be a very time consuming process.
- Tips For Large Caches

Invest First in a Small Area for all Scales

Make sure relevant data is visible

Select Appropriate Scales

Do all the Areas Require all Scales?

Build Smaller Area Caches First

Reset your Full Extent

Specify Extent Areas (Cities)

Use the Update Cache Method

Create Scales First to Represent All

Select Scales Appropriate for each Area

1st level	1:16,000,000	1 tile
2nd level	1:8,000,000	4 tiles
3rd level	1:4,000,000	16 tiles
4th level	1:2,000,000	64 tiles
5th level	1:1,000,000	256 tiles
6th level	1:500,000	1,024 tiles
7th level	1:250,000	4,096 tiles
8th level	1:125,000	16,384 tiles
9th level	1:62,500	65,536 tiles
10th level	1:31,250	262,144 tiles

What type of cache should I create?

- Fused cache
 - Includes all layers in map in one “fused” image
 - Good performance
 - Can’t toggle layers on and off
- Multilayer cache
 - Can choose groups of layers to be cached separately
 - Performance decreases with number of layer groups
 - Can toggle layers on and off

Help

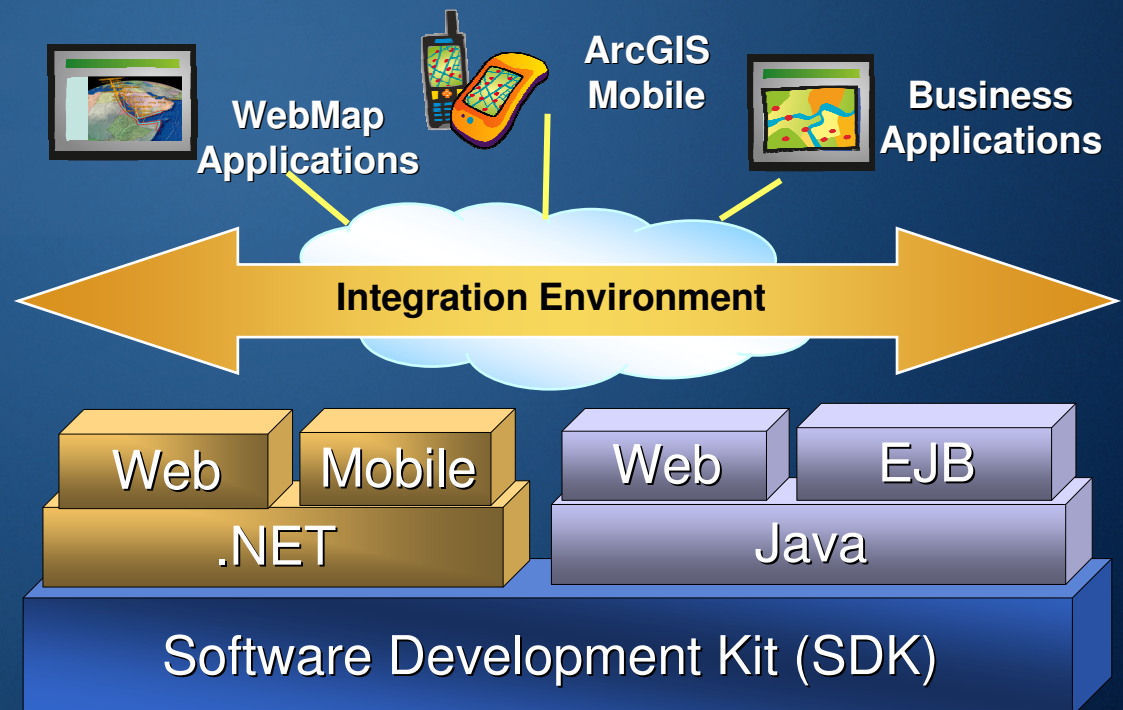
- Local Help
- <http://Blogs.esri.com>
- <http://webhelp.esri.com>



The .NET Development Environment in ArcGIS Server 9.2

ArcGIS Server 9.2: Software Development Kit

- Build and deploy web & enterprise geospatial applications and services
- Productivity boost with out-of-the-box IDE integration
- Software Development Kit (SDK) includes :
 - .NET components
 - Web ADF
 - Mobile ADF
 - Java components
 - Web ADF
 - EJB ADF



Getting Started - Developing Web Applications with ArcGIS Server 9.2

Web Application Developer Framework (Web ADF)

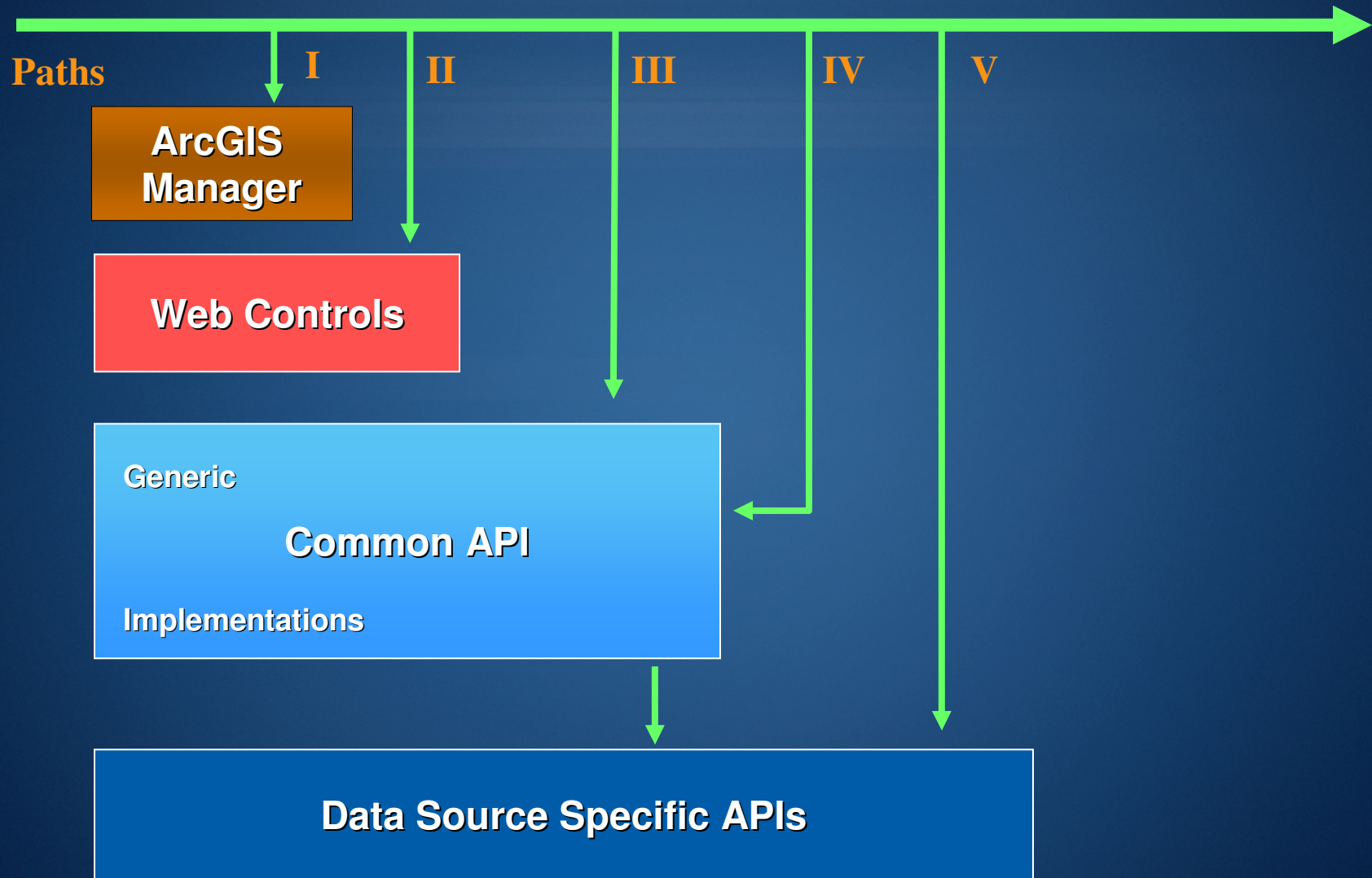
- ArcGIS Server Manager
 - Build Web Applications
- Supports multiple data sources
 - ArcGIS Server, ArcIMS, ArcWeb, WMS custom, etc.
- Multi-source controls
 - Map image blending
 - AJAX enabled
- Task Framework and Web Tasks

Developing Web Applications

- Use Visual Studio 2005 – 2.0 Framework
 - OR Visual Studio Web Developer Express (FREE!!)
- Choices on how to get started developing
 - Beginner*: Edit the web application created in the Website Manager
 - Intermediate*: Start from a template integrated into the development environment – Web Controls
 - Advanced*: Common Data Source API, Data Source Specific API

Paths of Development

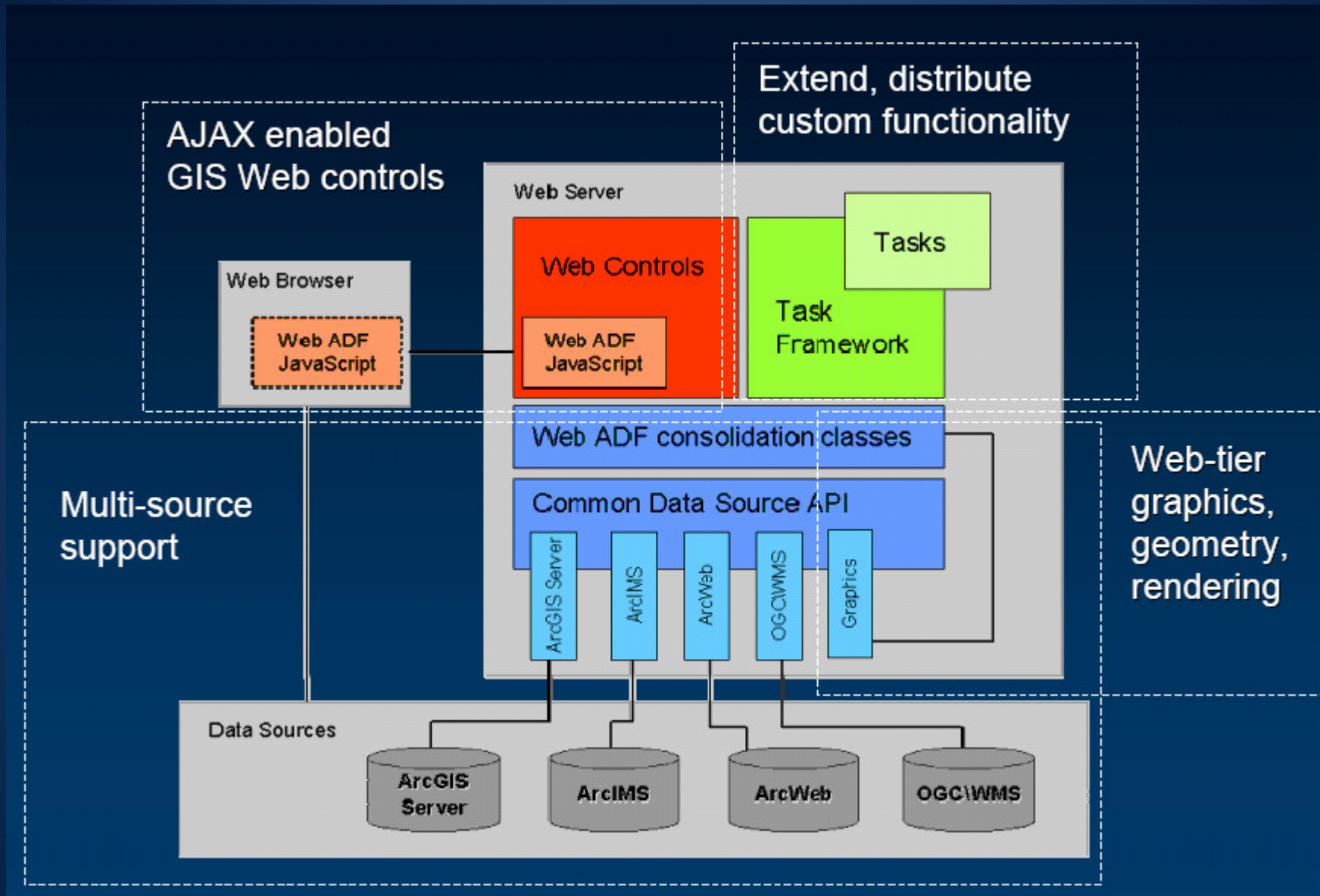
- Increasing complexity and functionality



DEMO

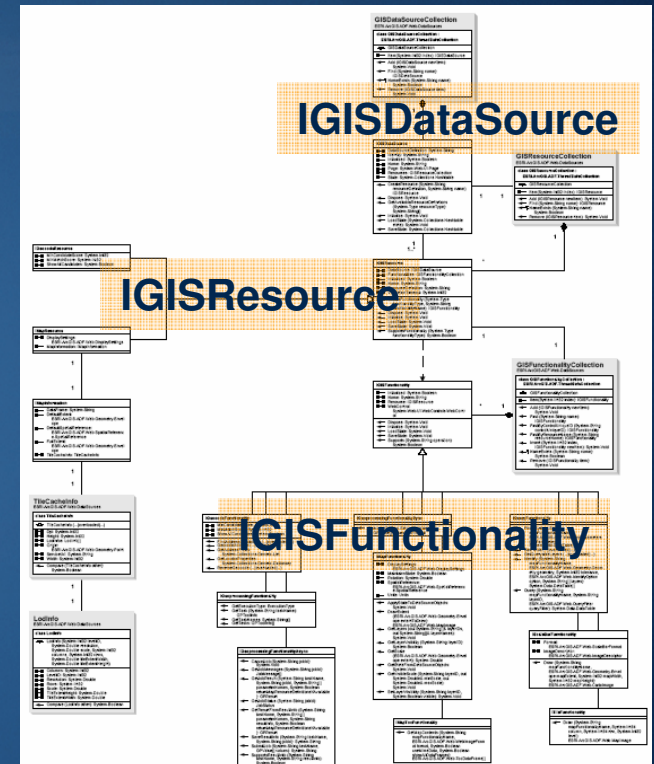
ArcGIS Server Manager
ADF Template
ADF Controls

The Common Data Source API



Common Data Source Cont.

- All data sources implement a common set of interfaces
 - ArcGIS Server
 - ArcIMS
 - ArcWeb Services
 - Open Geospatial Consortium (OGC)
 - Graphics
- Provides a generic way for Web controls to access functionality
 - Draw a map, query a layer, geocode
- See ESRI.ArcGIS.ADF.Web.DataSources
- Common Functionalities



.NET Classes that reside on the Web Server

NOT ArcObjects!

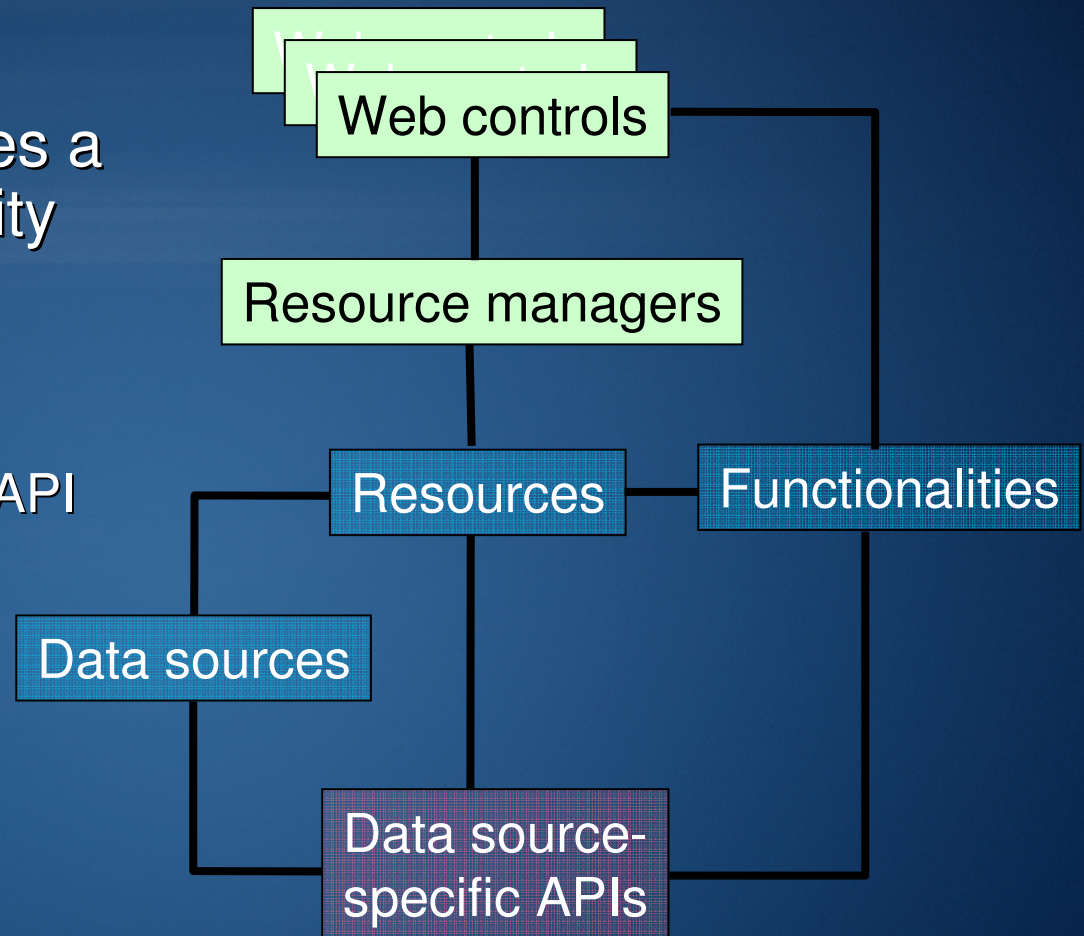
Advantages of the Common Data Source API

- Removes business and GIS logic from Web controls
- Easy to program against different data sources
- Possible to implement your own custom data sources

Users will be able to develop most of their needs from the Common Datasource API

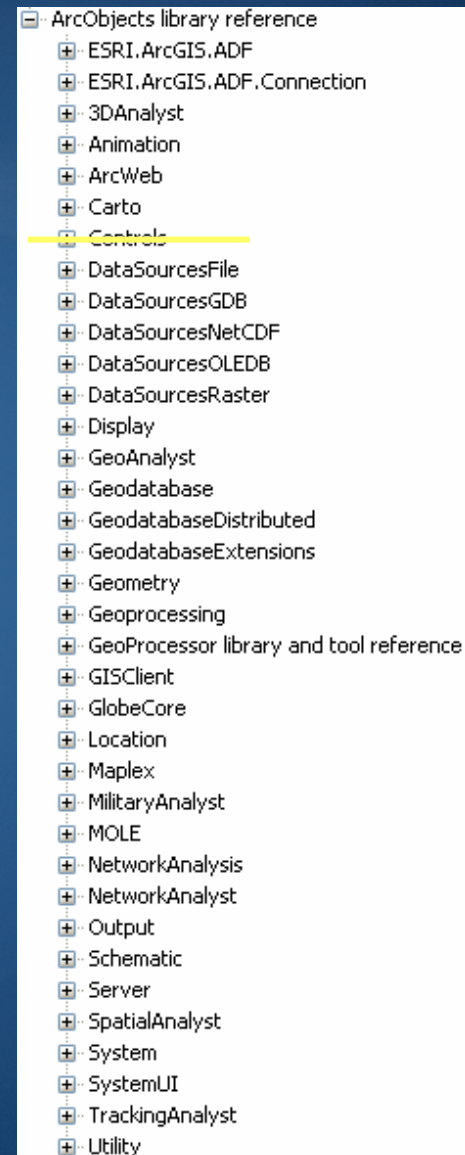
Data Source-specific APIs

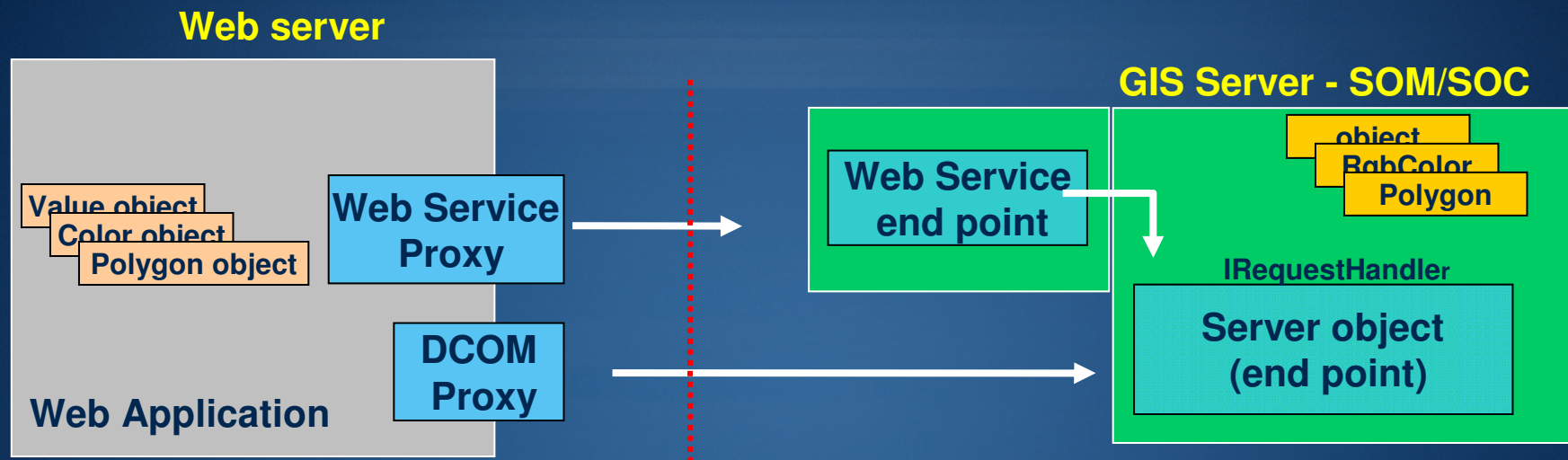
- Each data source exposes a different set of functionality
 - ArcGIS Server
 - SOAP, ArcObjects
 - ArcIMS – AXL
 - ArcWeb Services – SOAP API
- What does this mean?
 - Many other data source-specific classes available
 - More business/GIS logic
 - Different APIs use different communication protocols
 - Requires different programming patterns for each data source



ArcGIS Server Local: Available ArcObjects

- Most of the Engine libraries
- Capabilities
 - Display
 - Symbolization
 - Analysis
 - Query
 - Data access
 - Editing
 - Output
- Also access extensions

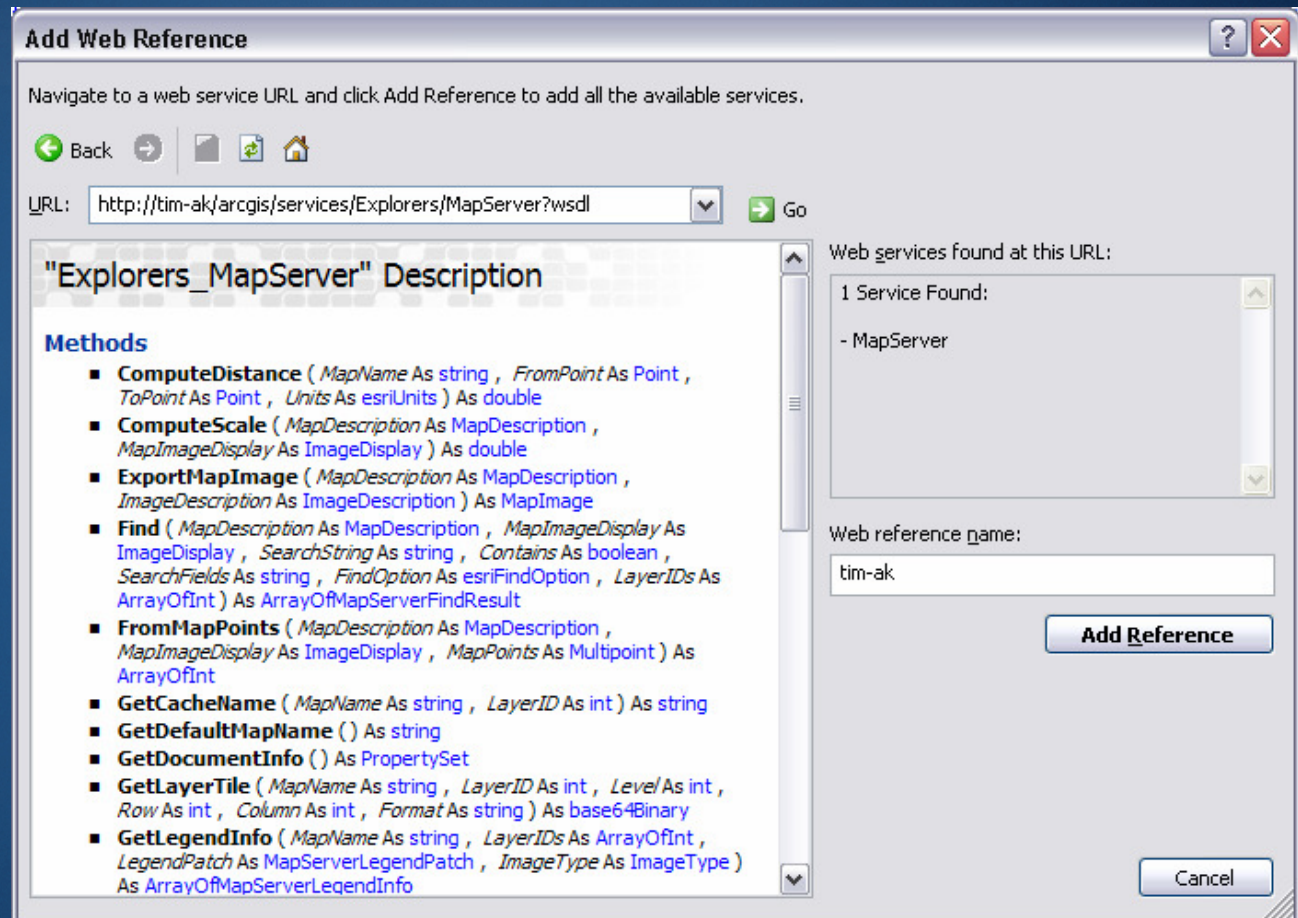




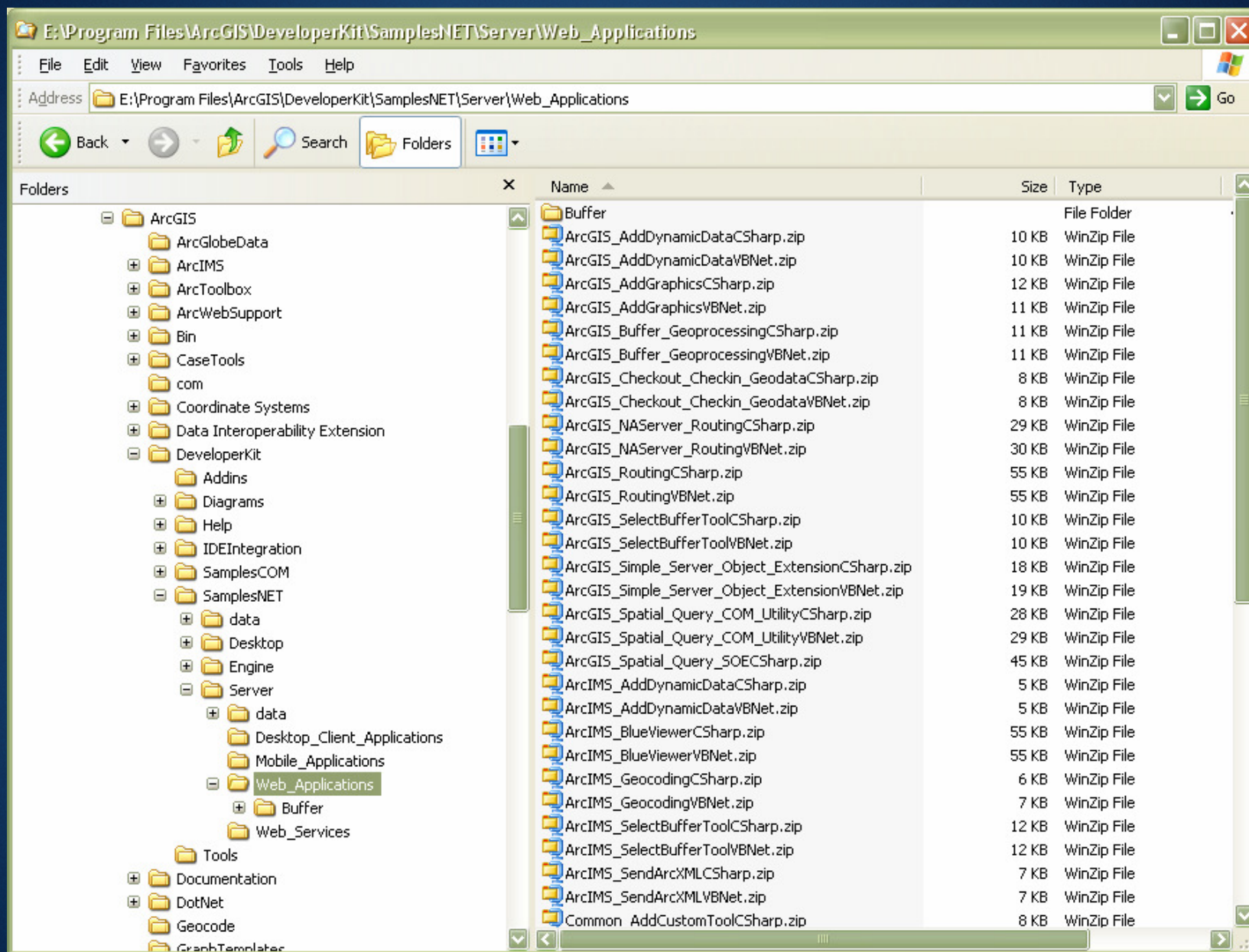
SOAP API – Web ADF or NOT!

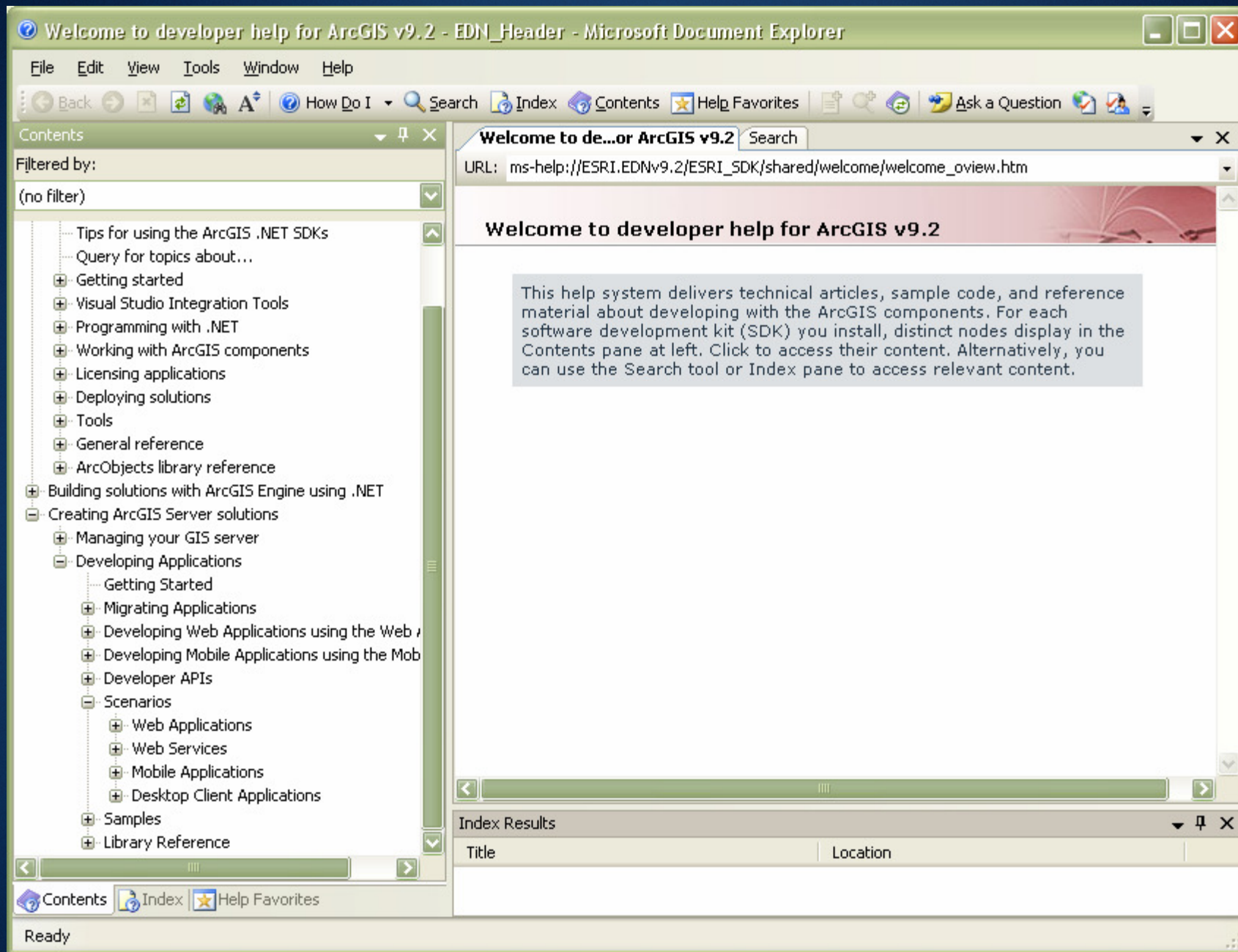
MapServer, Geocoder, Geoprocessing API's
All ArcGIS Services have a Web Service End Point

[ServiceCatalog](#)
[MapServer](#)
[GeocodeServer](#)
[GPSTServer](#)
[GeoDataServer](#)
[GlobeServer](#)
[NAServer](#)



SDK Developer samples





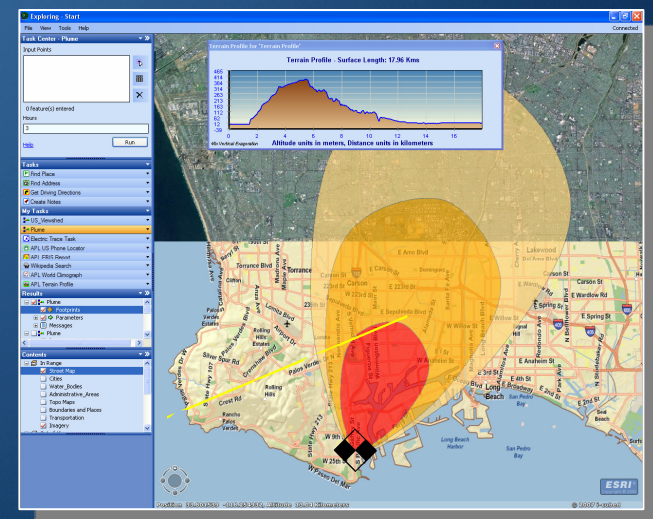


Configuration and Customizing ArcGIS Explorer

What is ArcGIS Explorer?

- A client for ArcGIS Server, offering an easy way to deliver access to GIS content and capabilities.
- A free, lightweight, easy to use desktop application that can access, integrate, and utilize GIS services, geographic content, and other web services.

***More than an exploration tool,
it's a way to deliver and
publish ArcGIS capabilities
to your users***



ArcGIS Explorer

Key Points

- An integrated part of the ArcGIS System
- Supports 2D and 3D (Globe) services
- Can fuse multiple services
 - ArcGIS Server, ArcIMS, WMS, Any web service
- Many base maps available (ArcGIS Online)
- Local content support
 - Shapefile, file GDB, KML, imagery, text, .csv, ...
- Tasks
- Can be centrally managed
- Free to download, free for any use

How do you get ArcGIS Explorer?

- Part of ArcGIS
 - Installed and configured with ArcGIS Server
- Download from ESRI Web site
 - <http://www.esri.com/arcgisexplorer>
 - <http://www.arcgisexplorer.com>



Demo

Introduction to ArcGIS Explorer

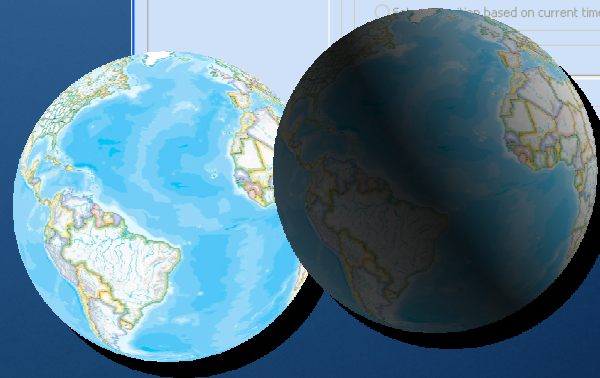
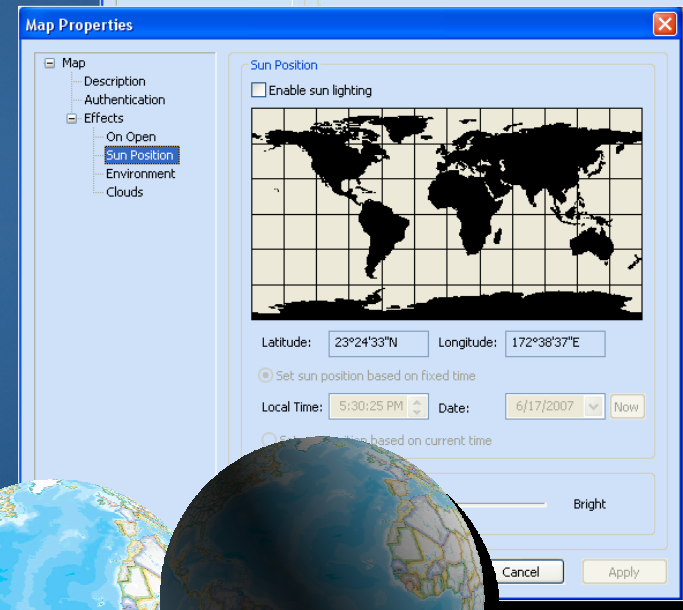
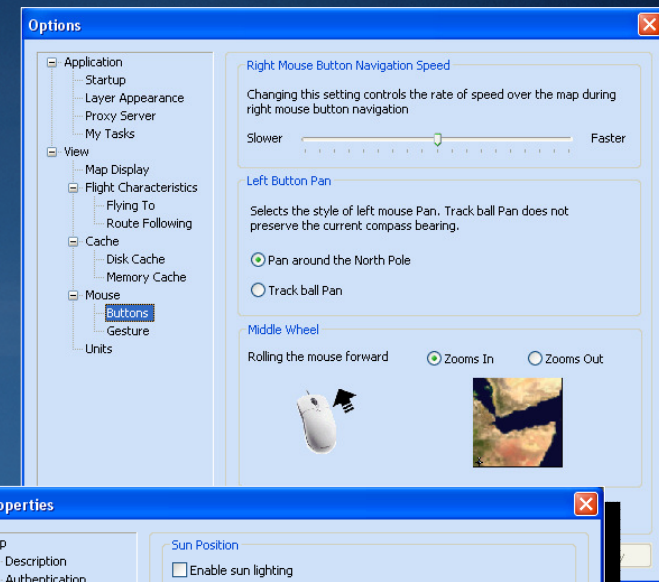


ArcGIS Explorer Configuration

ArcGIS Explorer

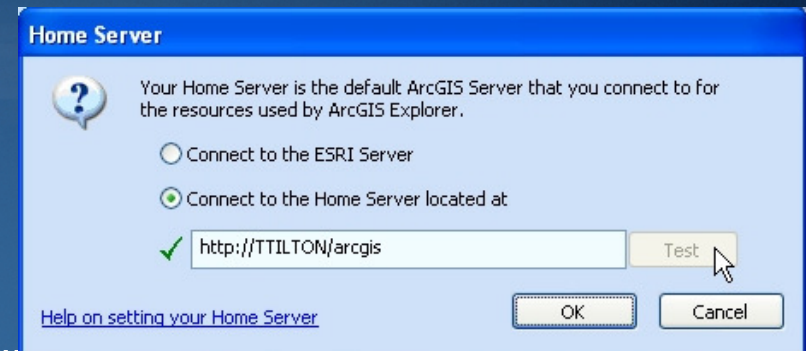
User Preferences

- Look and Feel of Globe
 - Sun shading, clouds, halo, stars...
- Map Display and Control
 - Navigation, flight control
 - Units
- Mouse
 - Button configuration, gestures, speed...
- Cache and memory management



Setting the ArcGIS Explorer home server

- Home server configures settings for Explorer clients
 - Default map
 - Ability to open or save documents
 - Appearance (skin file)
 - Others
- Modify settings for your home server
 - Edit configuration settings in E2Config.xml file
- ESRI is the default home server
 - To change, click File > Set Home Server
 - Define home server using the appropriate URL



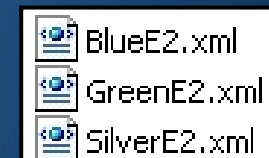
C:\Documents and Settings\tweisenburger\Application Data\ESRI\ArcGIS Explorer

ArcGIS Explorer customization: skin files

C:\Inetpub\wwwroot\ArcGIS\Explorer

Based on Home Server Connection

- Define basic appearance
 - Default skin for your install
 - For users of your home server
 - Font and background colors user interface images, ToolTips, etc.
- Three skins: blue, green, silver
 - XML file and associated images
 - Applied according to current operating system settings
- Edit default skins
 - Define different colors
 - Point to different images

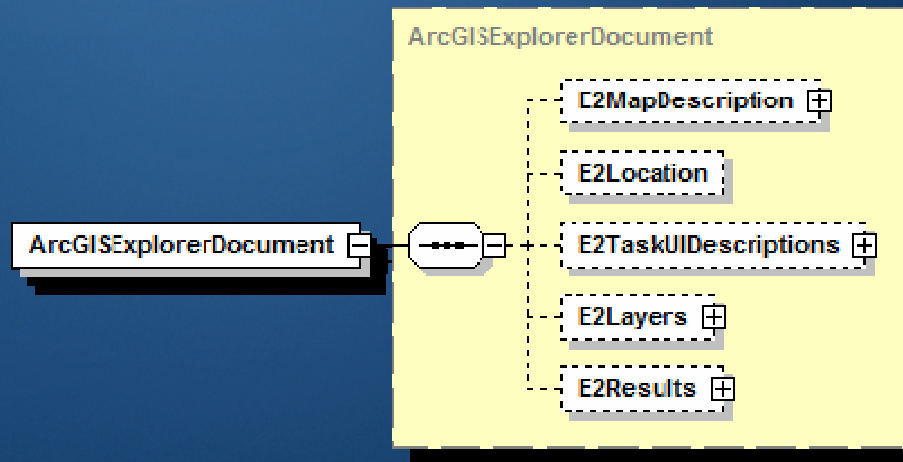


```
<BorderColor type="Color">RGB(0,0,128)</BorderColor>
<MenuFont type="Color">RGB(255,0,0)</MenuFont>
<TaskFont type="Color">RGB(255,0,0)</TaskFont>
<MenuHotFont type="Color">RGB(221,221,255)</MenuHotFont>
<TaskHotFont type="Color">RGB(0,0,128)</TaskHotFont>
<GeneralBackgroundColor type="Color">RGB(207,225,249)</GeneralBackgroundColor>
<MenuAndTaskStartColor type="Color">RGB(158,190,245)</MenuAndTaskStartColor>
```


NMF Files

- NMFs are how you share maps
 - put on website or fileshare
- Can contain
 - Maps and settings
 - Layers
 - Results
 - Tasks
- XML
 - published schema
 - allows programmatic creation, editing

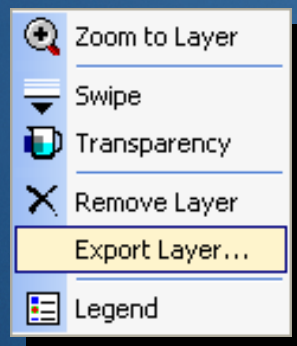
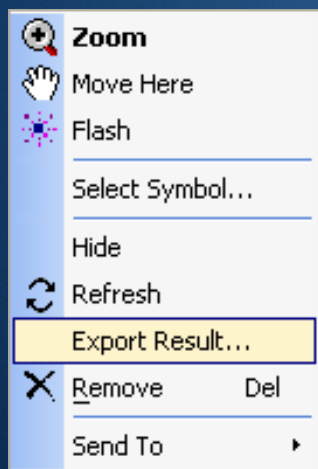
Name	Size	Type
DefaultMap.nmf	20 KB	ArcGIS Explorer Map
LayerLABuildings.nmf	1 KB	ArcGIS Explorer Map
LayerPremiumServices.nmf	6 KB	ArcGIS Explorer Map
LocationESRI.nmf	1 KB	ArcGIS Explorer Map
Mars_Grayscale.nmf	2 KB	ArcGIS Explorer Map
RouteRedlandsToSanDiego.nmf	218 KB	ArcGIS Explorer Map
worldgrid.nmf	499 KB	ArcGIS Explorer Map



ArcGIS Explorer

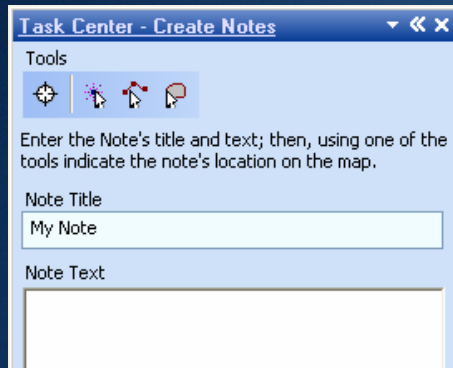
Maps, Tasks, Results

- A map (NMF) is a container for
 - Layers, Tasks, Results,
- XML

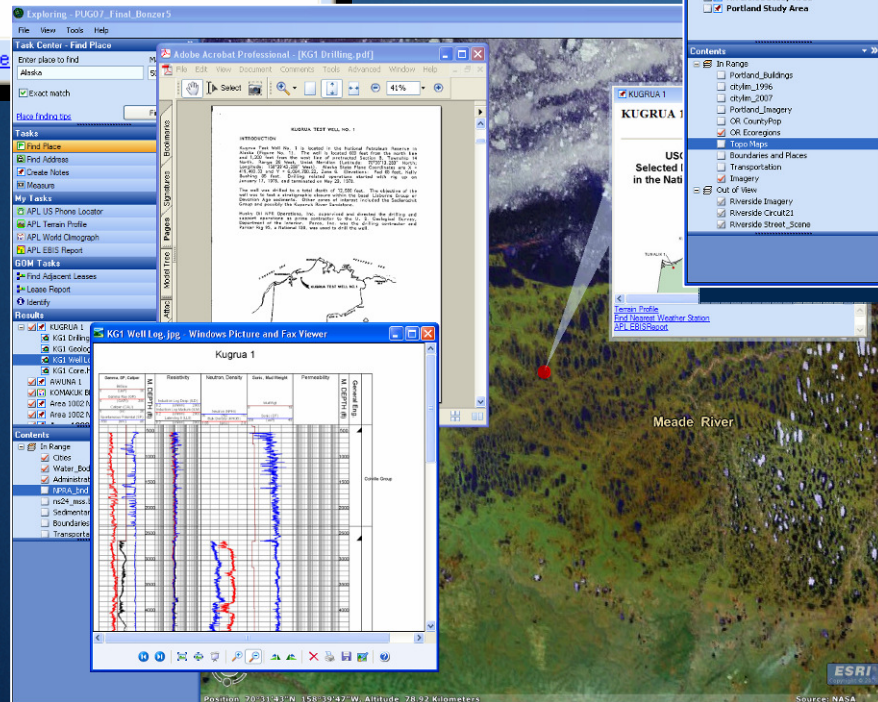


```
- <esri:ArcGISExplorerDocument
  xmlns:esri="http://www.esri.com/schemas/ArcGIS/9.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
- <E2MapDescription xsi:type="esri:E2MapDescription3D">
  <Name>PerfTest_July28</Name>
  <Description />
  <Copyright />
</E2MapDescription>
- <E2Location xsi:type="esri:E2Location3D">
- <Target xsi:type="esri:E2Point">
  - <Point xsi:type="esri:PointN">
    <X>-77.6469227857029</X>
    <Y>43.1961587650683</Y>
    <Z>-1.27247105994551E-08</Z>
  </Point>
</Target>
- <Observer xsi:type="esri:E2Point">
  - <Point xsi:type="esri:PointN">
    <X>-77.8044981879237</X>
    <Y>43.183867132051</Y>
    <Z>32039.9006352357</Z>
  </Point>
</Observer>
</E2Location>
```


ArcGIS Explorer Notes

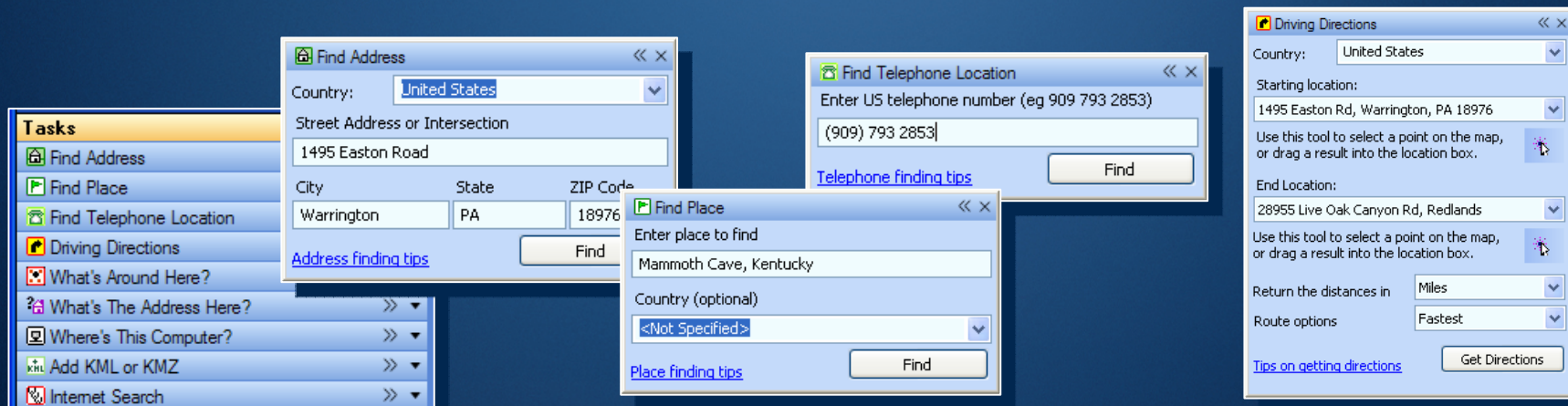


Creating Note



ArcGIS Explorer Tasks

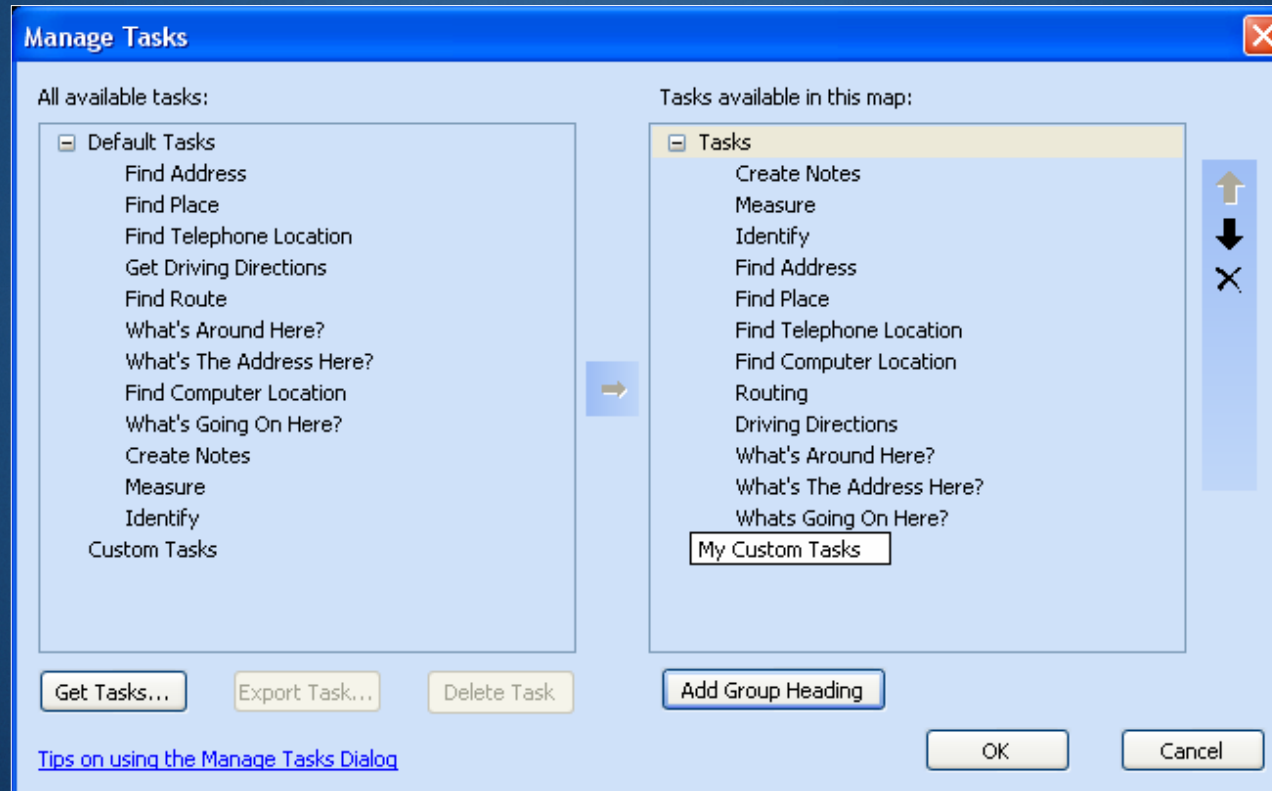
- Default Tasks
 - Powered by ArcGIS Online
- Custom Geoprocessing Tasks
 - Authored using ArcGIS Desktop and Published using ArcGIS Server
 - No programming necessary
- SDK for extending tasks, or building new tasks



ArcGIS Explorer Tasks

Task Manager

- Get, Export, Delete
- Arrange, Group



Providing Web GIS functionality in 3 steps

1. Author – create geoprocessing functionality

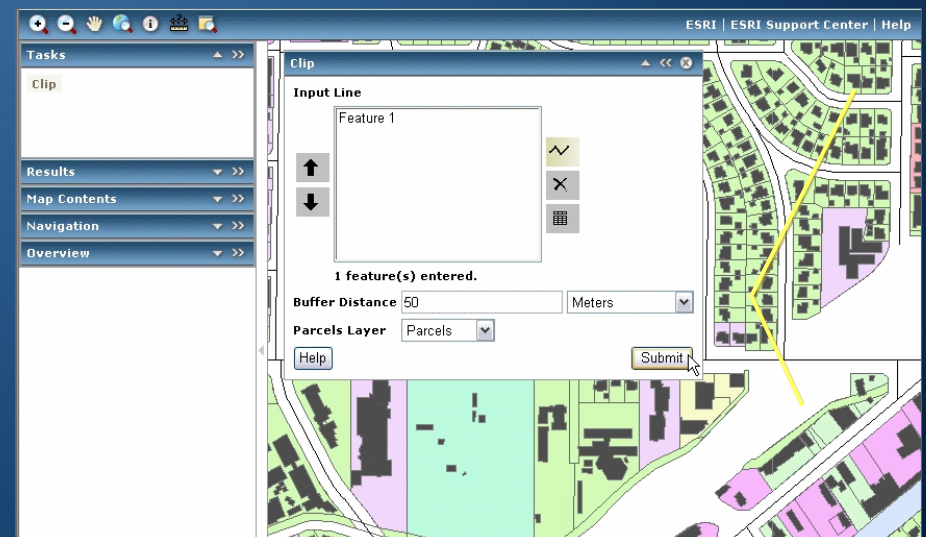
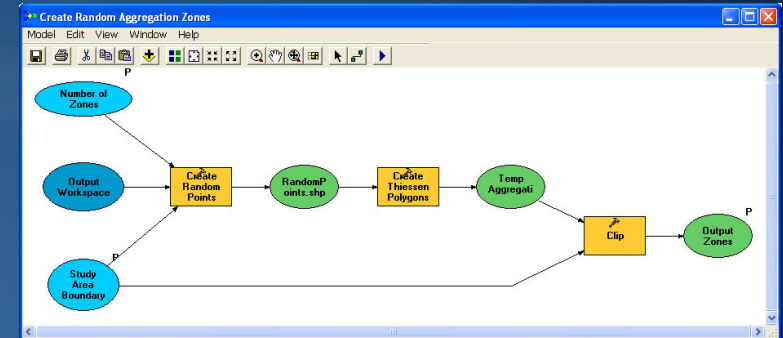
- Geoprocessing models

Publish – create the GIS service

- ## 2.
- Toolbox (.tbx)
 - Map document (.mxd)

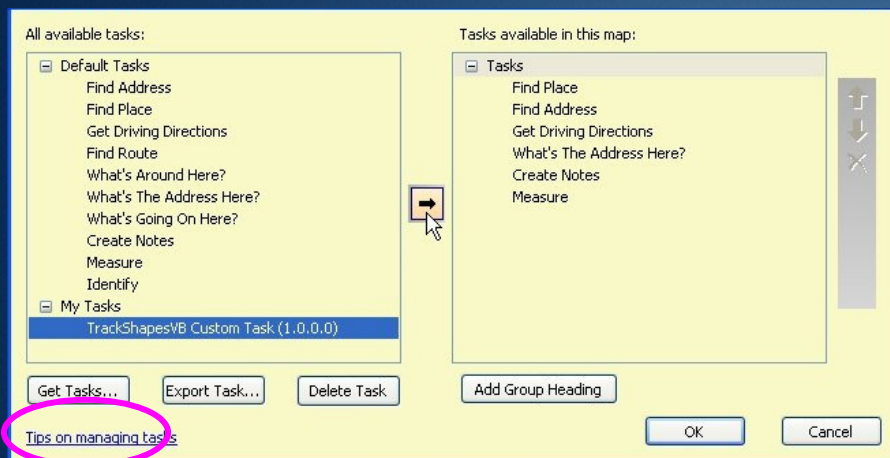
Consume – use the service

- ## 3.
- ArcGIS Desktop
 - ArcGIS Explorer
 - Web mapping application
 - Custom clients

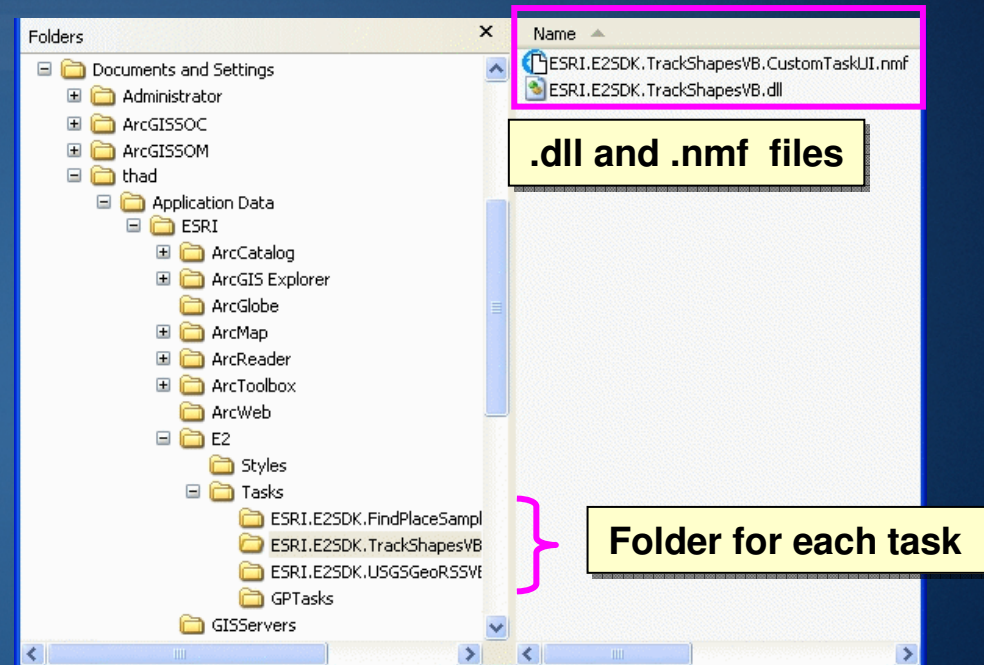


Adding custom tasks to ArcGIS Explorer

- Stored in Tasks folder in user profile directory
 - Read when Explorer starts
 - Appear in Task Manager

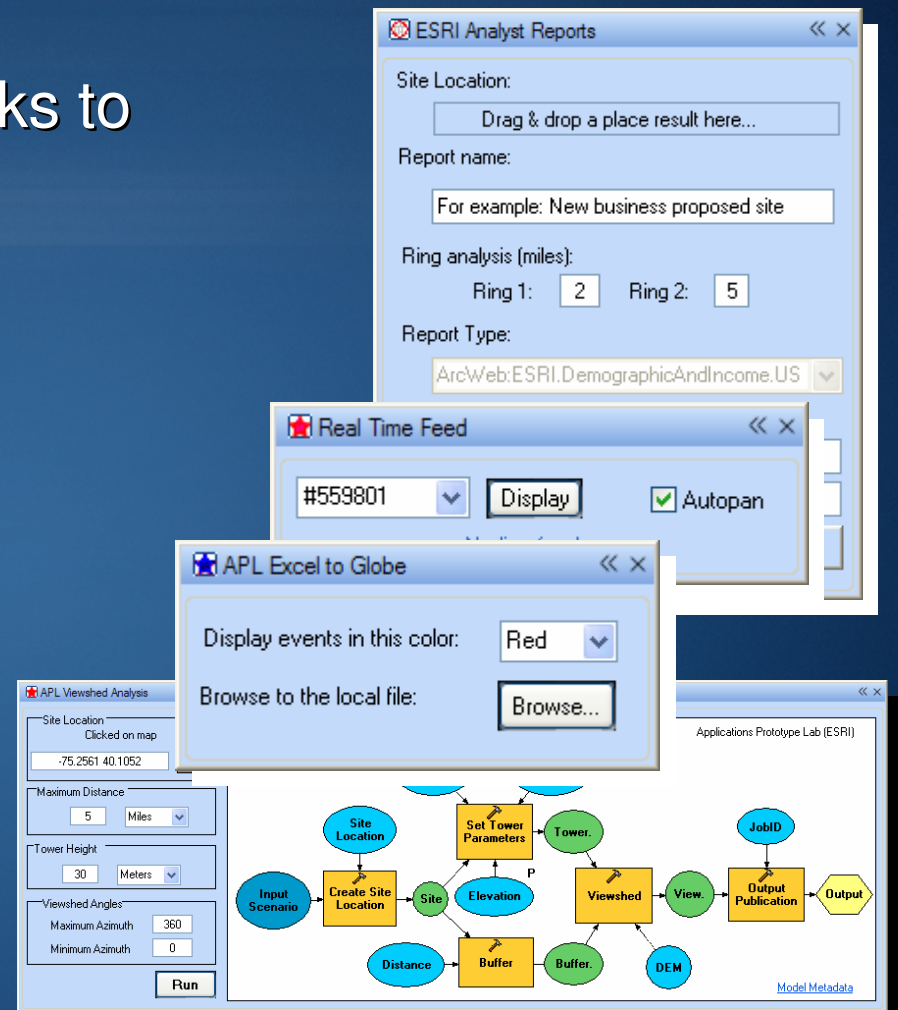


- Tasks can be added
 - By navigating to an .nmf file
 - Opening a map containing task information
 - From a geoprocessing service



Custom Tasks for ArcGIS Explorer

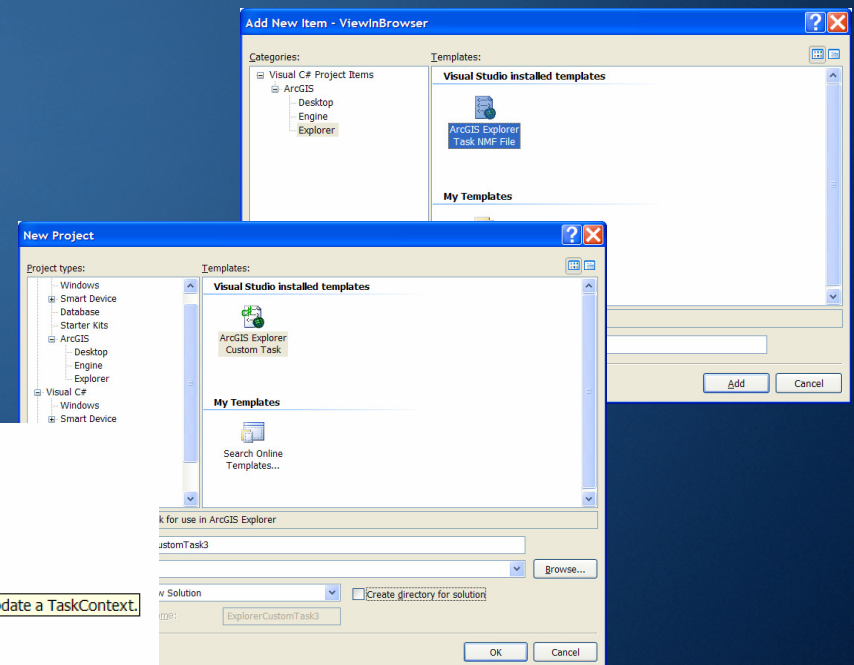
- Developers Create Custom Tasks to connect to new web services
- Can also
 - Refine existing user interface
 - Perform local operations



The Software Developer Kit

- ArcGIS Explorer .NET SDK provides resource to create custom tasks
 - Conceptual Documentation
 - Component Help
 - Visual Studio 2005 Integration
 - Object Model Diagram

```
class ViewInBrowserTask : Task
{
    public override void Execute(TaskContext taskContext)
    {
        Insert Snippet: ArcGIS Explorer >
        ParameterSet parameters();
        string url = parameters().ToString();
        Process.Start(url);
    }
}
```



E2API Classes

- Application and View classes
- Layer classes
- Task framework classes
- Result classes
- Geometry classes
- Feature classes



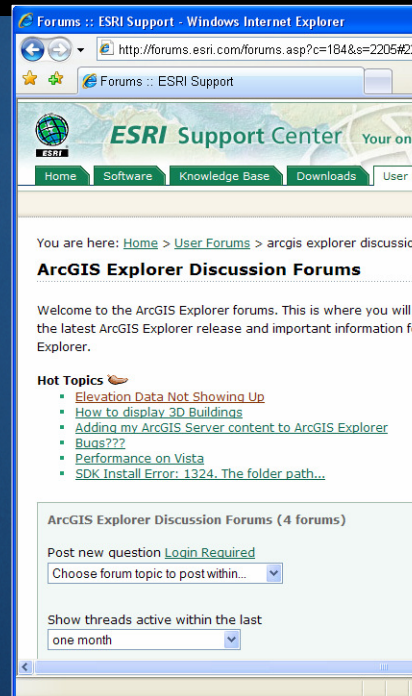
Demo

Default Tasks
Geoprocessing Tasks
Custom Tasks

ArcGIS Explorer Resource Center

Direct Access Resource Site

- Forum
- Blog
- More Fun...



- Tuning and configuring services
 - http://webhelp.esri.com/arcgisserver/9.2/dotNet/manager/publishing/tuning_services.htm
- Map Design Considerations for Dynamic Maps
 - http://webhelp.esri.com/arcgisserver/9.2/dotNet/manager/publishing/map_service.htm
- Creating Fill Symbols
 - http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=305&pid=297&topicname=Creating_fill_symbols
- Detailed view of all the web adf controls
 - http://edndoc.esri.com/arcobjects/9.2/NET_Server_Doc/developer/ADF/control_overview.htm
- Overview of programming with the common datasource API.
 - http://edndoc.esri.com/arcobjects/9.2/NET_Server_Doc/developer/ADF/resources.htm
- Overview of ArcIMS and ArcGIS Server data source specific APIs:
 - http://edndoc.esri.com/arcobjects/9.2/NET_Server_Doc/developer/developer_apis_overview.htm



Questions ?

Timothy R Weisenburger
tweisenburger@esri.com